# Quad 16-Bit,1 GSPS, TxDAC+ Digital-to-Analog Converter

Data Sheet **AD9148**

## FEATURES

- Single-carrier W-CDMA ACLR = 80 dBc at 150 MHz IF
- Channel-to-channel isolation > 90 dB
- Analog output
  - Adjustable 8.7 mA to 31.7 mA
  - $R_L$ = 25 Ω to 50 Ω
- Novel 2×, 4×, and 8× interpolator eases data interface
- On-chip fine complex NCO allows carrier placement anywhere in DAC bandwidth
- High performance, low noise PLL clock multiplier
- Multiple chip synchronization interface
- Programmable digital inverse sinc filter
- Auxiliary DACs allow for offset control
- Gain DACs allow for I and Q gain matching
- Programmable I and Q phase compensation
- Digital gain control
- Flexible LVDS digital I/F supports 32- or 16-bit bus width
- 196-ball CSP_BGA, 12 mm × 12 mm

## APPLICATIONS

- Wireless infrastructure
  - LTE, TD-SCDMA, WiMAX, W-CDMA, CDMA2000, GSM
- MIMO/transmit diversity
- Digital high or low IF synthesis

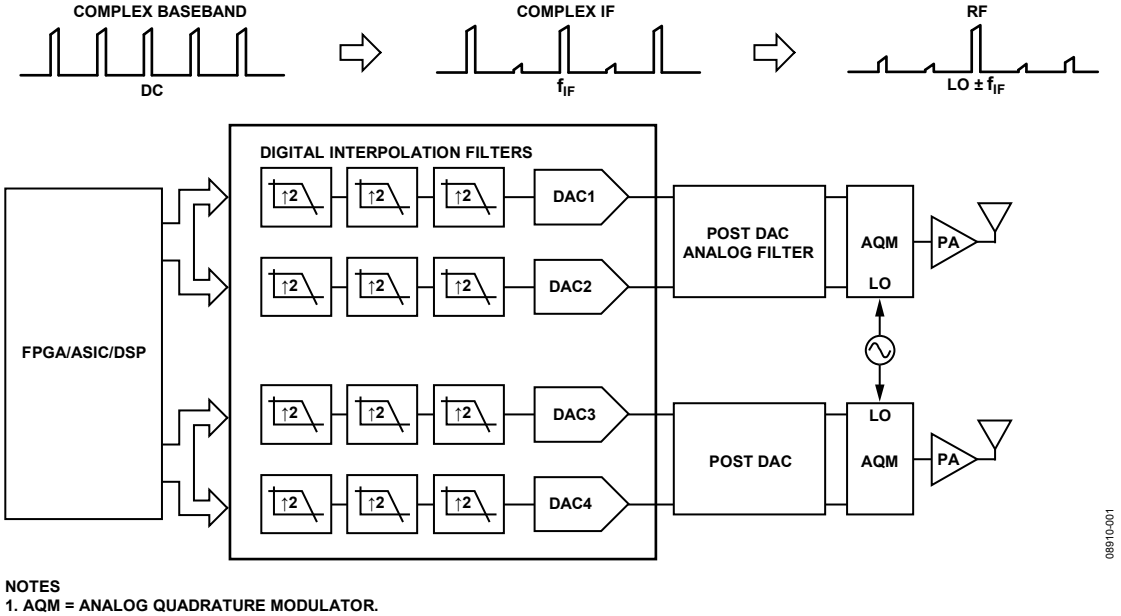
## GENERAL DESCRIPTION

The AD9148 is a quad, 16-bit, high dynamic range, digital-to-analog converter (DAC) that provides a sample rate of 1000 MSPS. This device includes features optimized for direct conversion transmit applications, including gain, phase, and offset compensation. The DAC outputs are optimized to interface seamlessly with analog quadrature modulators such as the ADL5371/ADL5372/ADL5373/ADL5374/ADL5375. A serial peripheral interface (SPI) is provided for programming of the internal device parameters. Full-scale output current can be programmed over a range of 8.7 mA to 31.7 mA. The device operates from 1.8 V and 3.3 V supplies for a total power consumption of 3 W at the maximum sample rate. The AD9148 is enclosed in a 196-ball chip scale package ball grid array with the option of an attached heat spreader.

## PRODUCT HIGHLIGHTS

1. Low noise and intermodulation distortion (IMD) enable high quality synthesis of wideband signals from baseband to high intermediate frequencies.
2. A proprietary DAC output switching technique enhances dynamic performance.
3. The current outputs are easily configured for various single-ended or differential circuit topologies.
4. The LVDS data input interface includes FIFO to ease input timing.

## TYPICAL SIGNAL CHAIN




NOTES
1. AQM = ANALOG QUADRATURE MODULATOR.

*Figure 1.*

**Rev. A**
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113 ©2010–2011 Analog Devices, Inc. All rights reserved.

# TABLE OF CONTENTS


Features ... 1
Applications ... 1
General Description ... 1
Product Highlights ... 1
Typical Signal Chain ... 1
Revision History ... 3
Functional Block Diagram ... 4
Specifications ... 5
DC Specifications ... 5
Input/Output Signal Specifications ... 6
Digital Input Data Timing Specifications ... 7
AC Specifications ... 8
Absolute Maximum Ratings ... 9
Thermal Resistance ... 9
Maximum Safe Power Dissipation ... 9
ESD Caution ... 9
Pin Configurations and Function Descriptions ... 11
Typical Performance Characteristics ... 15
Terminology ... 21
Serial Peripheral Interface ... 22
General Operation of the Serial Interface ... 22
Data Format ... 22
SPI Pin Descriptions ... 22
SPI Options ... 23
SPI Register Map ... 24
SPI Register Descriptions ... 26
Input Data Ports ... 40
Dual-Port Mode ... 40
Single-Port Mode ... 40
Byte Mode ... 41
Data Interface Options ... 41
Recommended Frame Input Bias Circuitry ... 41
FIFO Operation ... 42
Synchronizing and Resetting the FIFO ... 43
Monitoring the FIFO Status ... 44
Device Synchronization ... 45
Synchronizing Multiple Devices ... 45
Synchronization with Clock Multiplication ... 45
Synchronization with Direct Clocking ... 47
Additional Synchronization Features ... 48
Interface Timing ... 49
Digital Data Path ... 50
Premodulation ... 50
Programmable Inverse Sinc Filter ... 50
Interpolation Filters ... 51
Fine Modulation ... 54
Quadrature Phase Correction ... 55
DC Offset Correction ... 55
Digital Gain Control ... 55
Clock Generation ... 56
DAC Input Clock Configurations ... 56
Driving the CLK_x and REFCLK_x Inputs ... 56
Direct Clocking ... 56
Clock Multiplication ... 57
Analog Outputs ... 59
Transmit DAC Operation ... 59
Auxiliary DAC Operation ... 60
Interfacing to Modulators ... 61
Device Power Dissipation ... 63
Temperature Sensor ... 65
Interrupt Request Operation ... 66
Interrupt Service Routine ... 66
Interface Timing Validation ... 67
SED Operation ... 67
SED Example ... 67
Example Start-Up Routine ... 68
Derived PLL Settings ... 68
Derived NCO Settings ... 68
Start-Up Sequence ... 68
Device Verification Sequence ... 68
Outline Dimensions ... 69
Ordering Guide ... 70

## REVISION HISTORY

**9/11—Rev. 0 to Rev. A**

Changes to General Description Section ..... 1
Deleted Input High Voltage, $V_{IA}$ or $V_{IB}$ Parameter, Table 2, Input Low Voltage, $V_{IA}$ or $V_{IB}$ Parameter, Table 2, and Note 2, Table 2; Renumbered Sequentially ..... 6
Added Input Voltage Range, $V_{IA}$ or $V_{IB}$ Parameter, Table 2 ..... 6
Changes to Table 10 ..... 13
Changes to Figure 41 and Figure 42 ..... 23
Changes to 0x1E Addr, Table 12 ..... 24
Deleted 0x74 Row, Table 12 ..... 26
Changes to PLL Control 2, Table 13 ..... 29
Changes to HB3 Control, 1E, Bit 7 Row, Table 13 ..... 32
Deleted LVDS Pad Ctrl Row, Table 13 ..... 38
Deleted Frame Input Levels Section and Table 15 ..... 41
Added Recommended Frame Input Bias Circuitry Section and Figure 45; Renumbered Sequentially ..... 41
Changes to Timing Optimization Section ..... 48
Added Table 15 ..... 48
Changes to Filter Implementation Section ..... 50
Changes to Figure 74 ..... 57
Deleted Test Access Port Section, Table 27, Figure 92, and Table 28 ..... 68
Changes to Start-Up Sequence Section ..... 68
Deleted Figure 93 ..... 69
Deleted Table 29 ..... 70

**6/10—Revision 0: Initial Version**

# FUNCTIONAL BLOCK DIAGRAM

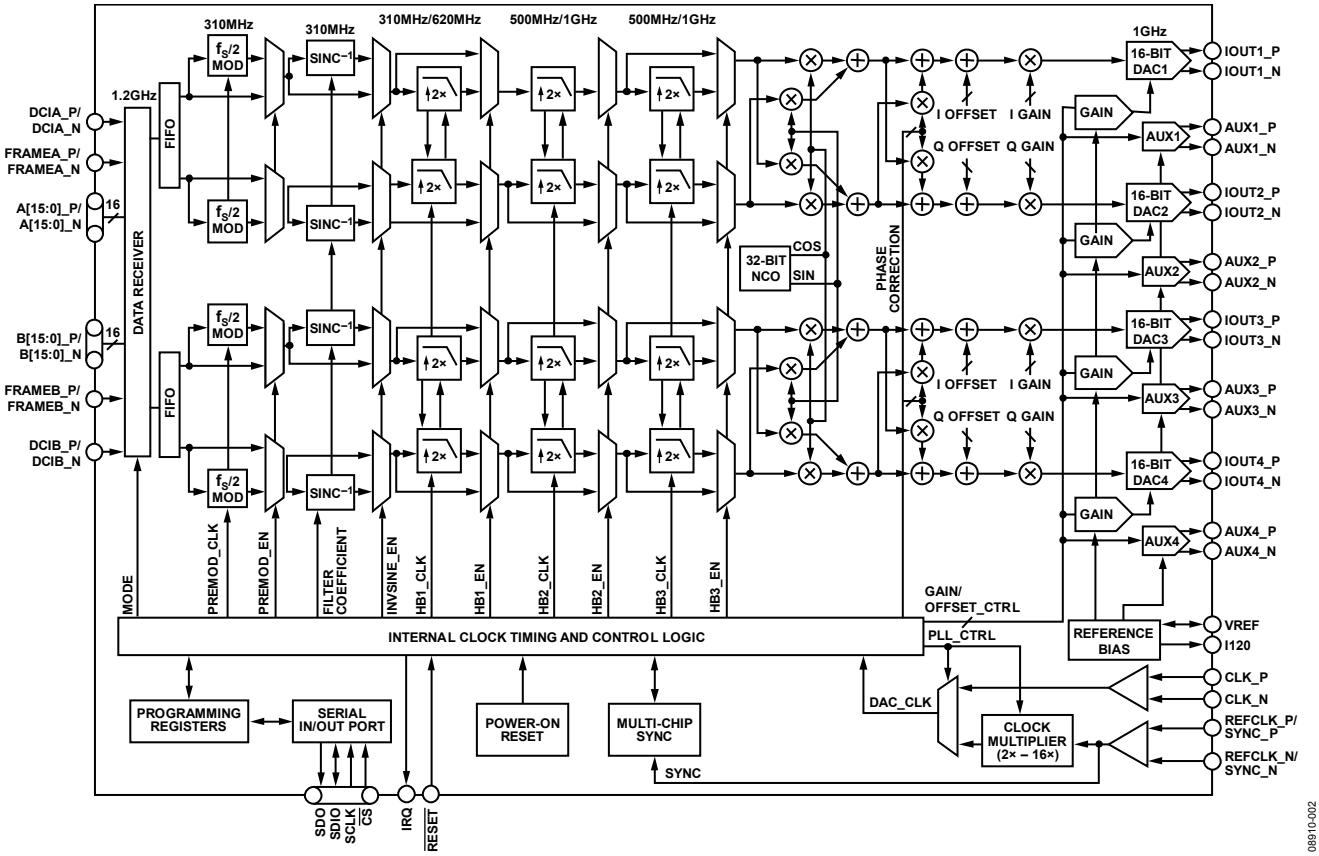



*Figure 2.*

# SPECIFICATIONS

## DC SPECIFICATIONS

$T_{MIN}$ to $T_{MAX}$, AVDD33 = 3.3 V, IOVDD = 3.3 V, DVDD18 = 1.8 V, CVDD18 = 1.8 V, $I_{OUTFS}$ = 20 mA, maximum sample rate, unless otherwise noted.

**Table 1.**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| RESOLUTION | | 16 | | Bits |
| ACCURACY | | | | |
| Differential Nonlinearity (DNL) | | ±2.1 | | LSB |
| Integral Nonlinearity (INL) | | ±3.7 | | LSB |
| MAIN DAC OUTPUTS | | | | |
| Offset Error | | ±0.001 | | % FSR |
| Gain Error (with Internal Reference) | | ±2 | | % FSR |
| Full-Scale Output Current[1] | 8.66 | 20.2 | 31.66 | mA |
| Output Compliance Range | −1.0 | | +1.0 | V |
| Output Resistance | | 10 | | MΩ |
| Gain DAC Monotonicity | | Guaranteed | | |
| Settling Time to Within ±0.5 LSB | | 20 | | ns |
| TEMPERATURE DRIFT | | | | |
| Main DAC Offset | | 0.04 | | ppm/°C |
| Main DAC Gain | | 100 | | ppm/°C |
| Reference Voltage | | 30 | | ppm/°C |
| REFERENCE | | | | |
| Internal Reference Voltage | | 1.2 | | V |
| Output Resistance | | 5 | | kΩ |
| ANALOG SUPPLY VOLTAGES | | | | |
| AVDD33 | 3.13 | 3.3 | 3.47 | V |
| CVDD18 | 1.71 | 1.8 | 1.89 | V |
| DIGITAL SUPPLY VOLTAGES | | | | |
| IOVDD | 1.71 | 1.8/3.3 | 3.47 | V |
| DVDD18 | 1.71 | 1.8 | 1.89 | V |
| POWER CONSUMPTION (NCO OFF, PLL DISABLED, AND $SINC^{-1}$ FILTER BYPASSED, UNLESS OTHERWISE NOTED) | | | | |
| 1 × Mode, $f_{DAC}$ = 300 MSPS, $f_{INTERFACE}$ = 600 MSPS | | 0.79 | | W |
| 2 × Mode, $f_{DAC}$ = 500 MSPS, $f_{INTERFACE}$ = 500 MSPS | | 1.49 | | W |
| 4 × Mode, $f_{DAC}$ = 800 MSPS, $f_{INTERFACE}$ = 400 MSPS | | 2.18 | | W |
| 4 × Mode, $f_{DAC}$ = 800 MSPS, $f_{INTERFACE}$ = 400 MSPS, NCO On | | 2.47 | | W |
| 4 × Mode, $f_{DAC}$ = 800 MSPS, $f_{INTERFACE}$ = 400 MSPS, PLL Enabled | | 2.26 | | W |
| 4 × Mode, $f_{DAC}$ = 800 MSPS, $f_{INTERFACE}$ = 400 MSPS, $Sinc^{-1}$ Filter Enabled | | 2.44 | | W |
| 8 × Mode, $f_{DAC}$ = 800 MSPS, $f_{INTERFACE}$ = 200 MSPS | | 2.01 | 2.16 | W |
| AVDD33 | | 368 | 373 | mW |
| CVDD18 | | 261 | 280 | mW |
| IOVDD | | 0.8 | 1.6 | mW |
| DVDD18 | | 1377 | 1504 | mW |
| Power-Down Mode | | 1 | 12 | mW |
| OPERATING RANGE | −40 | +25 | +85 | °C |

[1] Based on a 10 kΩ external resistor.

## INPUT/OUTPUT SIGNAL SPECIFICATIONS

$T_{MIN}$ to $T_{MAX}$, AVDD33 = 3.3 V, IOVDD = 3.3 V, DVDD18 = 1.8 V, CVDD18 = 1.8 V, $I_{OUTFS}$ = 20 mA, maximum sample rate, unless otherwise noted. LVDS driver and receiver are compliant to the IEEE-1596 reduced range link, unless otherwise noted.

**Table 2.**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| CMOS INPUT LOGIC LEVEL (SCLK, SDIO, $\overline{CS}$, $\overline{RESET}$, TMS, TDI, TCK) | | | | |
| Input $V_{IN}$ Logic High (IOVDD = 1.8 V) | 1.2 | | | V |
| Input $V_{IN}$ Logic High (IOVDD = 3.3 V) | 2.0 | | | V |
| Input $V_{IN}$ Logic Low (IOVDD = 1.8 V) | | | 0.6 | V |
| Input $V_{IN}$ Logic Low (IOVDD = 3.3 V) | | | 0.8 | V |
| CMOS OUTPUT LOGIC LEVEL (SDIO, SDO, $\overline{IRQ}$, PLL_LOCK, TDO) | | | | |
| Output $V_{OUT}$ Logic High (IOVDD = 1.8 V) | 1.4 | | | V |
| Output $V_{OUT}$ Logic High (IOVDD = 3.3 V) | 2.4 | | | V |
| Output $V_{OUT}$ Logic Low (IOVDD = 1.8 V) | | | 0.4 | V |
| Output $V_{OUT}$ Logic Low (IOVDD = 3.3 V) | | | 0.4 | V |
| LVDS RECEIVER INPUTS (A[15:0]_x, B[15:0]_x, DCIA_x, DCIB_x) | | | | |
| Input Voltage Range, $V_{IA}$ or $V_{IB}$ | 825 | | 1575 | mV |
| Input Differential Threshold, $V_{IDTH}$ | −100 | | +100 | mV |
| Input Differential Hysteresis, $V_{IDTHH}$ to $V_{IDTHL}$ | | 20 | | mV |
| Receiver Differential Input Impedance, $R_{IN}$ | 80 | | 120 | Ω |
| LVDS Input Rate, $f_{INTERFACE}$ (See Table 4) | | | 1200 | MSPS |
| LVDS RECEIVER INPUTS (FRAMEA_x, FRAMEB_x) | | | | |
| Input Voltage Range, $V_{IA}$ or $V_{IB}$ | 825 | | 1575 | mV |
| DAC CLOCK INPUT (CLK_P, CLK_N) | | | | |
| Differential Peak-to-Peak Voltage | 100 | 500 | 2000 | mV |
| Common-Mode Voltage (Self-Biasing, AC-Coupled) | | 1.25 | | V |
| Maximum Clock Rate | 1000 | | | MSPS |
| REFERENCE CLOCK INPUT (REFCLK_x/SYNC_x) | | | | |
| Differential Peak-to-Peak Voltage | 100 | 500 | 2000 | mV |
| Common-Mode Voltage (Self-Biasing, AC-Coupled) | | 1.25 | | V |
| Maximum Clock Rate | 500 | | | MSPS |
| Minimum Clock Rate (PLL Enabled) | | | | |
| Loop Divider = /2 | | | 125 | MSPS |
| Loop Divider = /4 | | | 62.5 | MSPS |
| Loop Divider = /8 | | | 31.25 | MSPS |
| Loop Divider = /16 | | | 15.625 | MSPS |
| SERIAL PERIPHERAL INTERFACE | | | | |
| Maximum Clock Rate (SCLK) | 40 | | | MHz |
| Minimum Pulse Width High ($t_{PWH}$) | | | 12.5 | ns |
| Minimum Pulse Width Low ($t_{PWL}$) | | | 12.5 | ns |
| Set-Up Time, SDI to SCLK ($t_{DS}$) | 1.9 | | | ns |
| Hold Time, SDI to SCLK ($t_{DH}$) | 0.2 | | | ns |
| Data Valid, SDO to SCLK ($t_{DV}$) | 23 | | | ns |
| Setup time, $\overline{CS}$ to SCLK ($t_{DCSB}$) | | 1.4 | | ns |

## DIGITAL INPUT DATA TIMING SPECIFICATIONS

**Table 3.**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| LATENCY (DACCLK CYCLES) | | | | |
| 1× Interpolation (with or Without Coarse Modulation) | | 64 | | Cycles |
| 2× Interpolation (with or Without Coarse Modulation) | | 125 | | Cycles |
| 4× Interpolation (with or Without Coarse Modulation) | | 254 | | Cycles |
| 8× Interpolation (with or Without Coarse Modulation) | | 508 | | Cycles |
| Inverse Sinc (1× Interpolation) | | 10 | | Cycles |
| Inverse Sinc (2× Interpolation) | | 20 | | Cycles |
| Inverse Sinc (4× Interpolation) | | 40 | | Cycles |
| Inverse Sinc (8× Interpolation) | | 80 | | Cycles |
| Fine Modulation | | 12 | | Cycles |
| Power–Up Time | | 100 | | ms |

**Table 4. Maximum Rate**

| | Maximum Rate (MSPS) | | | | | |
|---|---|---|---|---|---|---|
| **Interface Mode** | $f_{INTERFACE}$ | $f_{DATA}$ | $f_{HB1}$ | $f_{HB2}$ | $f_{HB3}$ | $f_{DAC}$ |
| Dual Port Mode | 620 | 310 | 620 | 1000 | 1000 | 1000 |
| Single Port Mode or Byte Mode | 1200 | 300 | 600 | 1000 | 1000 | 1000 |

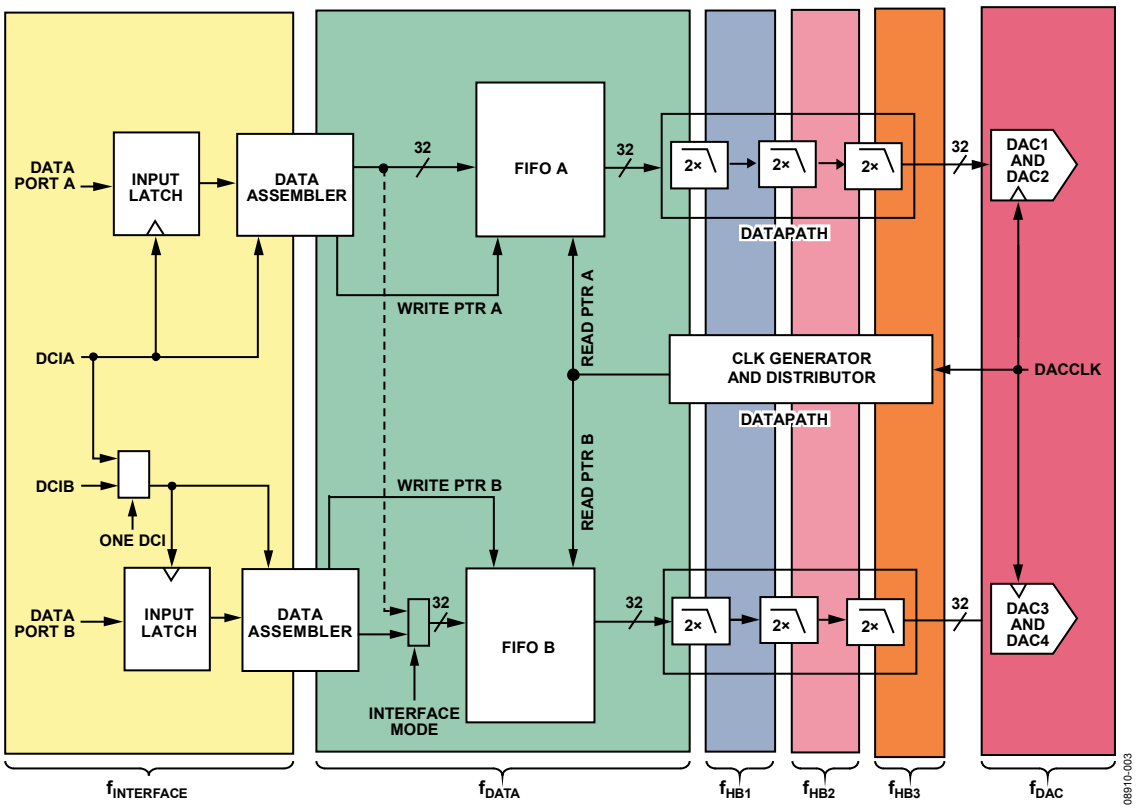



*Figure 3. Defining Maximum Rates*

## AC SPECIFICATIONS

$T_{MIN}$ to $T_{MAX}$, AVDD33 = 3.3 V, IOVDD = 3.3 V, DVDD18 = 1.8 V, CVDD18 = 1.8 V, $I_{OUTFS}$ = 20 mA, maximum sample rate, unless otherwise noted.

**Table 5.**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| SPURIOUS-FREE DYNAMIC RANGE (SFDR) | | | | |
| $f_{DAC}$ = 400 MSPS, $f_{OUT}$ = 80 MHz | | 72 | | dBc |
| $f_{DAC}$ = 600 MSPS, $f_{OUT}$ = 100 MHz | | 67 | | dBc |
| $f_{DAC}$ = 1000 MSPS, $f_{OUT}$ = 100 MHz | | 65 | | dBc |
| TWO-TONE INTERMODULATION DISTORTION (IMD) | | | | |
| $f_{DAC}$ = 400 MSPS, $f_{OUT}$ = 100 MHz | | 85 | | dBc |
| $f_{DAC}$ = 600 MSPS, $f_{OUT}$ = 120 MHz | | 82 | | dBc |
| $f_{DAC}$ = 1000 MSPS, $f_{OUT}$ = 150 MHz | | 76 | | dBc |
| NOISE SPECTRAL DENSITY (NSD) EIGHT-TONE, 500 kHz TONE SPACING | | | | |
| $f_{DAC}$ = 200 MSPS, $f_{OUT}$ = 80 MHz | | −160 | | dBm/Hz |
| $f_{DAC}$ = 400 MSPS, $f_{OUT}$ = 100 MHz | | −161 | | dBm/Hz |
| $f_{DAC}$ = 800 MSPS, $f_{OUT}$ = 100 MHz | | −162.5 | | dBm/Hz |
| W-CDMA ADJACENT CHANNEL LEAKAGE RATIO (ACLR), SINGLE CARRIER | | | | |
| $f_{DAC}$ = 737.28 MSPS, $f_{OUT}$ = 100 MHz, PLL Off | | −81 | | dBc |
| $f_{DAC}$ = 737.28 MSPS, $f_{OUT}$ = 100 MHz, PLL On | | −78 | | dBc |
| $f_{DAC}$ = 737.28 MSPS, $f_{OUT}$ = 200 MHz, PLL Off | | −79 | | dBc |
| $f_{DAC}$ = 737.28 MSPS, $f_{OUT}$ = 200 MHz, PLL On | | −72.5 | | dBc |
| W-CDMA ALTERNATE CHANNEL LEAKAGE RATIO, SINGLE CARRIER | | | | |
| $f_{DAC}$ = 737.28 MSPS, $f_{OUT}$ = 100 MHz, PLL Off | | −87 | | dBc |
| $f_{DAC}$ = 737.28 MSPS, $f_{OUT}$ = 100 MHz, PLL On | | −83 | | dBc |
| $f_{DAC}$ = 737.28 MSPS, $f_{OUT}$ = 200 MHz, PLL Off | | −84 | | dBc |
| $f_{DAC}$ = 737.28 MSPS, $f_{OUT}$ = 200 MHz, PLL On | | −80.5 | | dBc |

# ABSOLUTE MAXIMUM RATINGS

**Table 6.**

| Parameter | With Respect To | Rating |
|---|---|---|
| AVDD33, IOVDD | AGND, DGND, CGND | −0.3 V to +3.6 V |
| DVDD18, CVDD18 | AGND, DGND, CGND | −0.3 V to +2.10 V |
| AGND | DGND, CGND | −0.3 V to +0.3 V |
| DGND | AGND, CGND | −0.3 V to +0.3 V |
| CGND | AGND, DGND | −0.3 V to +0.3 V |
| I120, VREF | AGND | −0.3 V to AVDD33 + 0.3 V |
| IOUT1_P, IOUT1_N, IOUT2_P, IOUT2_N, IOUT3_P, IOUT3_N, IOUT4_P, IOUT4_N | AGND | −1.0 V to AVDD33 + 0.3 V |
| A15_P to A0_P, A15_N to A0_N, B15_P to B0_P, B15_N, B0_N | DGND | −0.3 V to DVDD18 + 0.3 V |
| DCIA_P, DCIA_N, FRAMEA_P, FRAMEA_N, DCIB_P, DCIB_N, FRAMEB_P, FRAMEB_N | DGND | −0.3 V to DVDD18+ 0.3 V |
| CLK_P, CLK_N, REFCLK_P, REFCLK_N | CGND | −0.3 V to CVDD18 + 0.3 V |
| CSB, SCLK, SDIO, SDO, $\overline{\text{TDO}}$, TDI, TCK, TMS, $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, PLL_LOCK | DGND | −0.3 V to IOVDD + 0.3 V |
| Junction Temperature | | 125°C |
| Storage Temperature Range | | −65°C to +150°C |

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## THERMAL RESISTANCE

Typical $\theta_{JA}$, $\theta_{JB}$, and $\theta_{JC}$ are specified vs. the number of PCB layers in still air for each package offering. Airflow increases heat dissipation effectively reducing $\theta_{JA}$ and $\theta_{JB}$.

**Table 7. Thermal Resistance**

| Package Type | $\theta_{JA}$ | $\theta_{JB}$ | $\theta_{JC}$ | Unit | Notes |
|---|---|---|---|---|---|
| 196-Ball CSP_BGA | 24.7 | 12.6 | 5.7 | °C/W | 4-layer board, 25 PCB vias |
| | 19.2 | 10.9 | 5.3 | °C/W | 8-layer board, 25 PCB vias |
| | 18.1 | 10.5 | 5.3 | °C/W | 10-layer board, 25 PCB vias |
| | 18.0 | 10.5 | 5.3 | °C/W | 12-layer board, 25 PCB vias |
| 196-Ball BGA_ED | 20.9 | 8.6 | 3.1 | °C/W | 4-layer board, 25 PCB vias |
| | 16.2 | 7.7 | 3.1 | °C/W | 8-layer board, 25 PCB vias |
| | 15.2 | 7.4 | 3.1 | °C/W | 10-layer board, 25 PCB vias |
| | 15.0 | 7.4 | 3.1 | °C/W | 12-layer board, 25 PCB vias |

## MAXIMUM SAFE POWER DISSIPATION

The maximum junction temperature for the AD9148 is 125°C. With the thermal resistance of the molded package (CSP_BGA) given for a 12 layer board, the maximum power that can be dissipated in this package can be calculated as

$$Power_{MAX} = \frac{(T_J - T_A)}{\theta_{JA}} = \frac{(125-85)}{18.0} = 2.22\ W$$

To increase the maximum power, the AD9148 is available in a second package option (BGA_ED), which includes a heat spreader on top of the package. Also, an external heat sink can be attached to the top of the AD9148 CSP_BGA package. The adjusted maximum power for each of these conditions is shown in Table 8.

With the thermal resistance of the heat spreader package (BGA_ED) given for a 12-layer board, the maximum power that can be dissipated in this package can be calculated as

$$Power_{MAX} = \frac{(T_J - T_A)}{\theta_{JA}} = \frac{(125-85)}{15.0} = 2.67\ W$$

To increase the maximum power, an external heat sink can be attached to the top of the AD9148 BGA_ED package. The adjusted maximum power for an external heat sink is shown in Table 8.

To aid in the selection of package, the maximum $f_{DAC}$ rate for a given power dissipation over several operating conditions is shown in Table 9. The maximum $f_{DAC}$ rate applies to all interpolation rates. Note that, if the programmable inverse sinc filter is enabled, the maximum $f_{DAC}$ rate specified in Table 9 decreases.

## ESD CAUTION

**ESD (electrostatic discharge) sensitive device.** Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

**Table 8. Thermal Resistance and Maximum Power**

| Package Type | $T_A$ (°C) | PCB | | | Case | $T_J$ (°C) | $\theta_{JA}$ (°C/W) | Maximum Power (W) |
|---|---|---|---|---|---|---|---|---|
| | | PCB Layers | PCB Vias | External Heat Sink[1] | | | | |
| 196-ball CSP_BGA | 85 | 12 | 25 | No | CSP_BGA | 125 | 18.0 | 2.22 |
| 196-ball CSP_BGA | 85 | 12 | 25 | Yes | CSP_BGA | 125 | 16.0 | 2.50 |
| 196-ball BGA_ED | 85 | 12 | 25 | No | BGA_ED | 125 | 15.0 | 2.67 |
| 196-ball BGA_ED | 85 | 12 | 25 | Yes | BGA_ED | 125 | 14.0 | 2.86 |

[1] Heat sink is used in the thermal model: 13 mm × 13 mm, 15 mm tall.

**Table 9. Power vs. $f_{DAC}$ Rate and Functionality**

| Maximum Power (W) | Package | Heat-Sink Combination[2] | Maximum $f_{DAC}$ (MSPS)[1] | | | |
|---|---|---|---|---|---|---|
| | | | Coarse Modulation | | Fine Modulation (NCO) | |
| | | | PLL Off | PLL On | PLL Off | PLL On |
| 2.22 | CSP_BGA | No | 820 | 740 | 695 | 630 |
| 2.50 | CSP_BGA | Yes | 950 | 875 | 810 | 740 |
| 2.67 | BGA_EP | No | 1000 | 945 | 870 | 810 |
| 2.86 | BGA_EP | Yes | 1000 | 1000 | 940 | 870 |

[1] Typical maximum $f_{DAC}$ rate with inverse sinc filter off.
[2] Heat sink is used in the thermal model: 13 mm × 13 mm, 15 mm tall.

# PIN CONFIGURATIONS AND FUNCTION DESCRIPTIONS

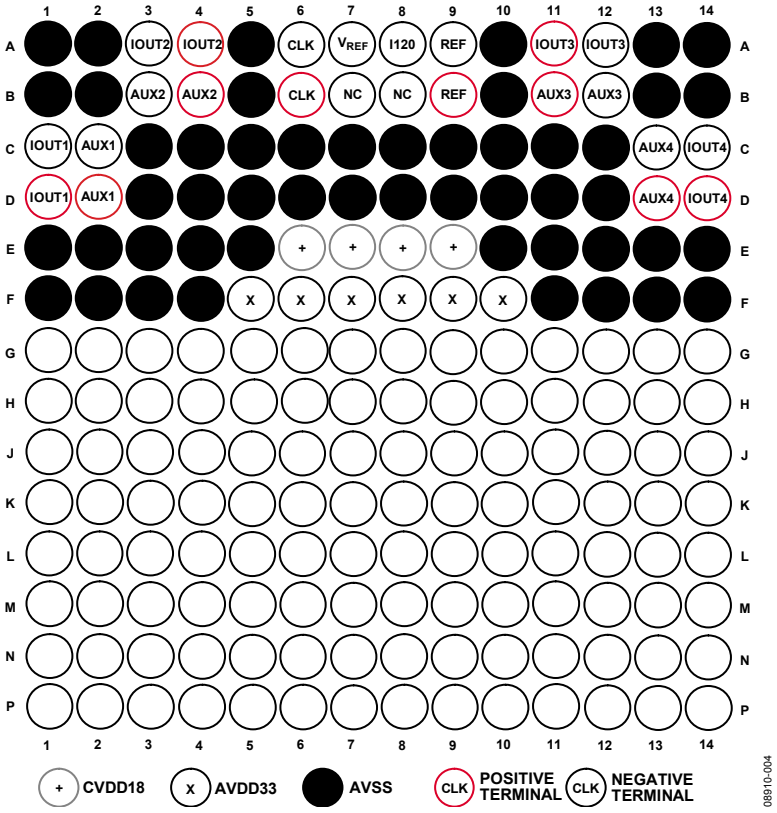

1 2 3 4 5 6 7 8 9 10 11 12 13 14
A IOUT2 IOUT2 CLK VREF I120 REF IOUT3 IOUT3 A
B AUX2 AUX2 CLK NC NC REF AUX3 AUX3 B
C IOUT1 AUX1 AUX4 IOUT4 C
D IOUT1 AUX1 AUX4 IOUT4 D
E + + + + E
F X X X X X X F
G G
H H
J J
K K
L L
M M
N N
P P
1 2 3 4 5 6 7 8 9 10 11 12 13 14
+ CVDD18
X AVDD33
AVSS
CLK POSITIVE TERMINAL
CLK NEGATIVE TERMINAL
08910-004

*Figure 4. Pin Configuration (Top View), Analog and Clock Domain Pins*

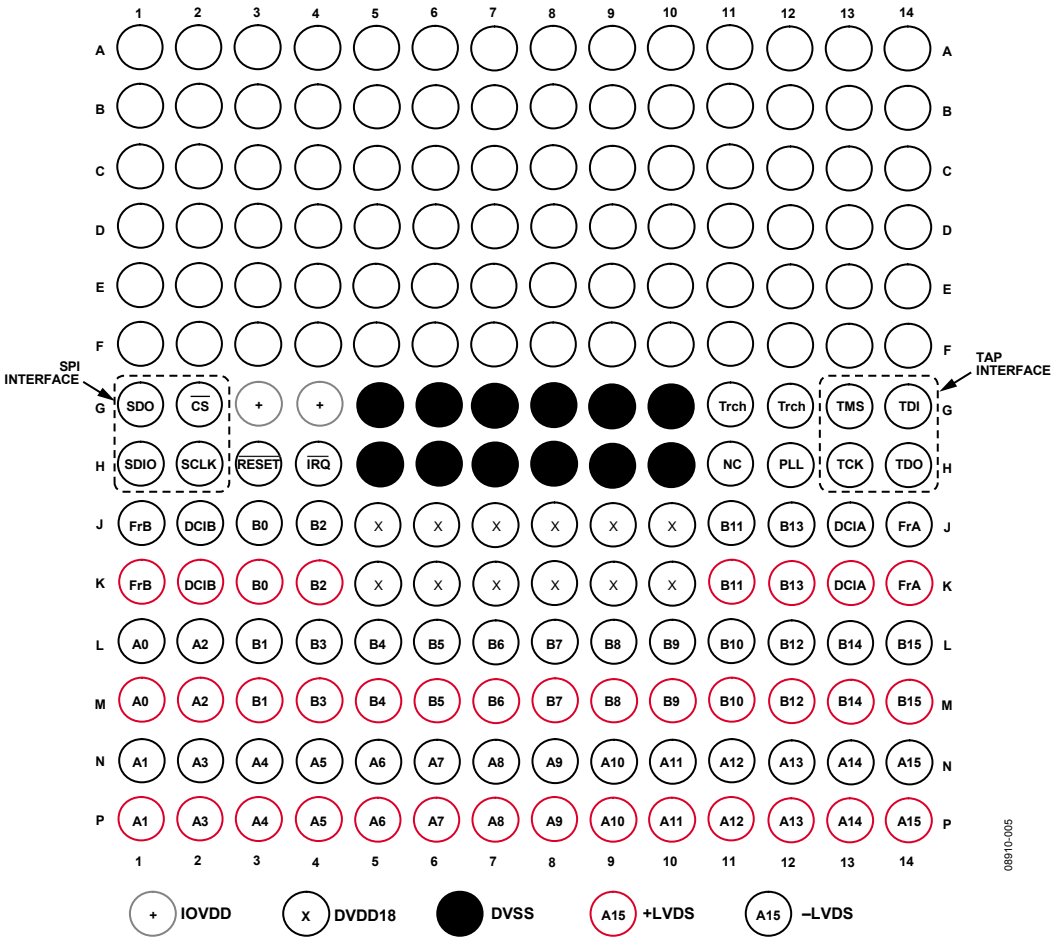



*Figure 5. Pin Configuration (Top View), Digital Domain Pins*

**Table 10. Pin Function Description**

| Pin No. | Mnemonic | Description |
|---|---|---|
| E6, E7, E8, E9 | CVDD18 | 1.8 V Clock Supply. |
| F5, F6, F7, F8, F9, F10 | AVDD33 | 3.3 V Analog Supply. |
| A1, A2, A5, A10, A13, A14, B1, B2, B5, B10, B13, B14, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, E1, E2, E3, E4, E5, E10, E11, E12, E13, E14, F1, F2, F3, F4, F11, F12, F13, F14 | AVSS | Analog Supply Ground. |
| G5, G6, G7, G8, G9, G10, H5, H6, H7, H8, H9, H10 | DVSS | Digital Supply Ground. |
| G3, G4 | IOVDD | Supply for Serial Ports (SPI and TAP), $\overline{\text{RESET}}$ and $\overline{\text{IRQ}}$. 1.8 V to 3.3 V can be supplied to these pins. |
| J5, J6, J7, J8, J9, J10, K5, K6, K7, K8, K9, K10 | DVDD18 | 1.8 V Digital Supply. |
| B7, B8, H11 | NC | No Connect. Do not connect to this pin. |
| C1 | IOUT1_N | DAC 1 Complementary Output Current. |
| D1 | IOUT1_P | DAC 1 Positive Output Current. |
| A3 | IOUT2_N | DAC 2 Complementary Output Current. |
| A4 | IOUT2_P | DAC 2 Positive Output Current. |
| A11 | IOUT3_P | DAC 3 Positive Output Current. |
| A12 | IOUT3_N | DAC 3 Complementary Output Current. |

| Pin No. | Mnemonic | Description |
|---|---|---|
| C14 | IOUT4_N | DAC 4 Complementary Output Current. |
| D14 | IOUT4_P | DAC 4 Positive Output Current. |
| C2 | AUX1_N | Auxiliary DAC 1 Complementary Output Current. |
| D2 | AUX1_P | Auxiliary DAC 1 Positive Output Current. |
| B3 | AUX2_N | Auxiliary DAC 2 Complementary Output Current. |
| B4 | AUX2_P | Auxiliary DAC 2 Positive Output Current. |
| B11 | AUX3_P | Auxiliary DAC 3 Positive Output Current. |
| B12 | AUX3_N | Auxiliary DAC 3 Complementary Output Current. |
| C13 | AUX4_N | Auxiliary DAC 4 Complementary Output Current. |
| D13 | AUX4_P | Auxiliary DAC 4 Positive Output Current. |
| A8 | I120 | Tie to analog ground via a 10 kΩ resistor to generate a 120 µA reference current. |
| A7 | VREF | Band Gap Voltage Reference I/O. Decouple to analog ground via a 0.1 µF capacitor. Output impedance is approximately 5 kΩ. |
| B6, A6 | CLK_P/CLK_N | Positive/Negative DAC Clock Input (CLK). |
| B9, A9 | REFCLK_P/REFCLK_N or SYNC_P/SYNC_N | PLL Reference Clock Input (REFCLK_x). This pin has a secondary function as a synchronization input (SYNC_x). |
| H4 | $\overline{\text{IRQ}}$ | Active Low Open-Drain Interrupt Request Output. Pull up to IOVDD with a 10 kΩ resistor. |
| H3 | $\overline{\text{RESET}}$ | An active low LVCMOS input resets the device. Pull up to IOVDD. |
| G1 | SDO | Serial Data Output for SPI. |
| G2 | $\overline{\text{CS}}$ | Active Low Chip Select for SPI. |
| H1 | SDIO | Serial Data Input/Output for SPI. |
| H2 | SCLK | Qualifying Clock Input for SPI. |
| G11, G12 | TRENCH | Connect this pin to VSS. |
| H12 | PLL_LOCK | Active High LVCMOS Output. It indicates the lock status of the PLL circuitry. |
| G13 | TMS | Reserved for Future Use. Connect to DVSS. |
| G14 | TDI | Reserved for Future Use. Connect to DVSS. |
| H13 | TCK | Reserved for Future Use. Connect to DVSS. |
| H14 | TDO | Reserved for Future Use. Leave unconnected. |
| M1, L1 | A0_P/A0_N | LVDS Data Input Pair, Port A (LSB). |
| P1, N1 | A1_P/A1_N | LVDS Data Input Pair, Port A. |
| M2, L2 | A2_P/A2_N | LVDS Data Input Pair, Port A. |
| P2, N2 | A3_P/A3_N | LVDS Data Input Pair, Port A. |
| P3, N3 | A4_P/A4_N | LVDS Data Input Pair, Port A. |
| P4, N4 | A5_P/A5_N | LVDS Data Input Pair, Port A. |
| P5, N5 | A6_P/A6_N | LVDS Data Input Pair, Port A. |
| P6, N6 | A7_P/A7_N | LVDS Data Input Pair, Port A. |
| P7, N7 | A8_P/A8_N | LVDS Data Input Pair, Port A. |
| P8, N8 | A9_P/A9_N | LVDS Data Input Pair, Port A. |
| P9, N9 | A10_P/A10_N | LVDS Data Input Pair, Port A. |
| P10, N10 | A11_P/A11_N | LVDS Data Input Pair, Port A. |
| P11, N11 | A12_P/A12_N | LVDS Data Input Pair, Port A. |
| P12, N12 | A13_P/A13_N | LVDS Data Input Pair, Port A. |
| P13, N13 | A14_P/A14_N | LVDS Data Input Pair, Port A. |
| P14, N14 | A15_P/A15_N | LVDS Data Input Pair, Port A (MSB). |
| K13, J13 | DCIA_P/DCIA_N | LVDS Data Clock Input Pair for Port A. |
| K14, J14 | FRAMEA_P/FRAMEA_N | LVDS Frame Input for Port A. Tie to LVDS logic low if not used. Recommended external bias circuit is shown in Figure 49. |
| K3, J3 | B0_P/B0_N | LVDS Data Input Pair, Port B (LSB). |
| M3, L3 | B1_P/B1_N | LVDS Data Input Pair, Port B. |
| K4, J4 | B2_P/B2_N | LVDS Data Input Pair, Port B. |
| M4, L4 | B3_P/B3_N | LVDS Data Input Pair, Port B. |
| M5, L5 | B4_P/B4_N | LVDS Data Input Pair, Port B |
| M6, L6 | B5_P/B5_N | LVDS Data Input Pair, Port B. |

| Pin No. | Mnemonic | Description |
|---|---|---|
| M7, L7 | B6_P/B6_N | LVDS Data Input Pair, Port B. |
| M8, L8 | B7_P/B7_N | LVDS Data Input Pair, Port B. |
| M9, L9 | B8_P/B8_N | LVDS Data Input Pair, Port B. |
| M10, L10 | B9_P/B9_N | LVDS Data Input Pair, Port B. |
| M11, L11 | B10_P/B10_N | LVDS Data Input Pair, Port B. |
| K11, J11 | B11_P/B11_N | LVDS Data Input Pair, Port B. |
| M12, L12 | B12_P/B12_N | LVDS Data Input Pair, Port B. |
| K12, J12 | B13_P/B13_N | LVDS Data Input Pair, Port B. |
| M13, L13 | B14_P/B14_N | LVDS Data Input Pair, Port B. |
| M14, L14 | B15_P/B15_N | LVDS Data Input Pair, Port B (MSB). |
| K2, J2 | DCIB_P/DCIB_N | LVDS Data Clock Input Pair for Port B. |
| K1, J1 | FRAMEB_P/FRAMEB_N | LVDS Frame Input for Port B. Tie to LVDS logic low if not used. Recommended external bias circuit is shown in Figure 49. |

# TYPICAL PERFORMANCE CHARACTERISTICS

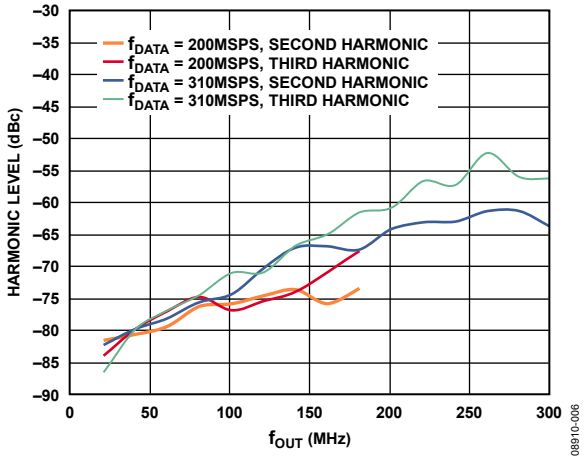



Figure 6. Harmonic Level vs. $f_{OUT}$ over $f_{DATA}$, 2× Interpolation, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

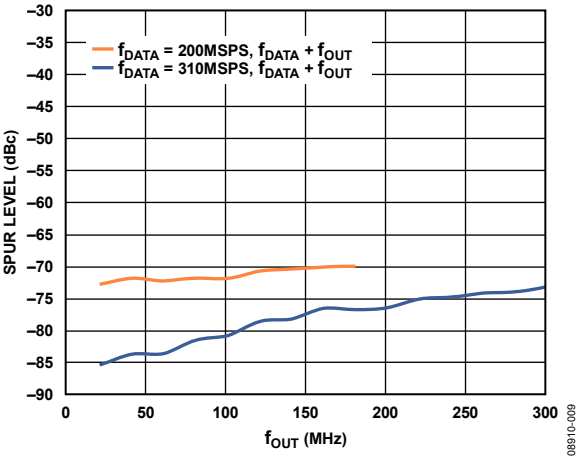



Figure 9. Highest Digital Spur vs. $f_{OUT}$ over $f_{DATA}$, 2× Interpolation, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

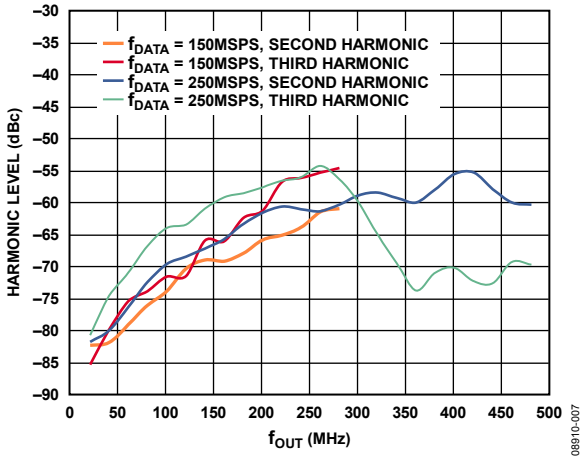



Figure 7. Harmonic Level vs. $f_{OUT}$ over $f_{DATA}$, 4× Interpolation, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

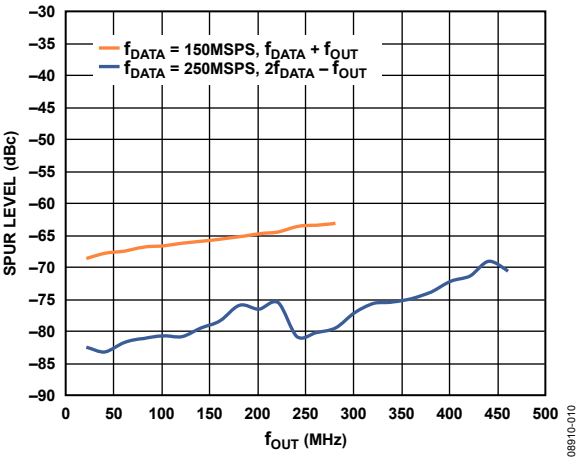



Figure 10. Highest Digital Spur vs. $f_{OUT}$ over $f_{DATA}$, 4× Interpolation, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

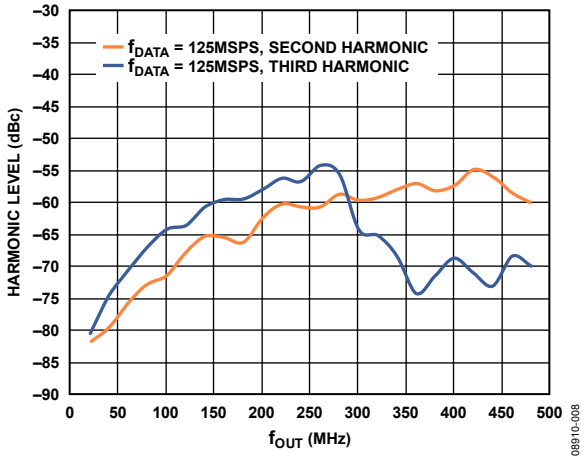



Figure 8. Harmonic Level vs. $f_{OUT}$, 8× Interpolation over $f_{DATA}$ = 125 MSPS, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

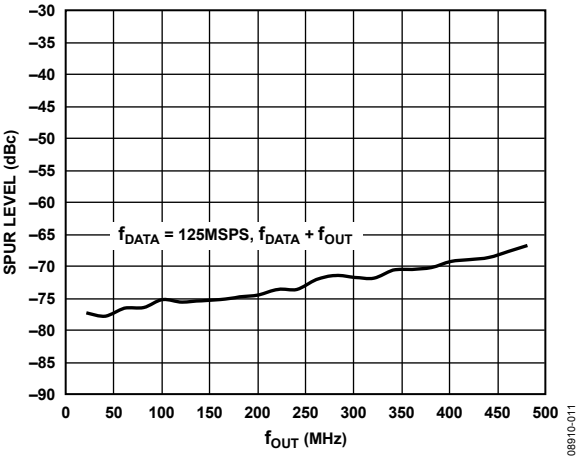



Figure 11. Highest Digital Spur vs. $f_{OUT}$, 8× Interpolation, $f_{DATA}$ = 125 MSPS, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

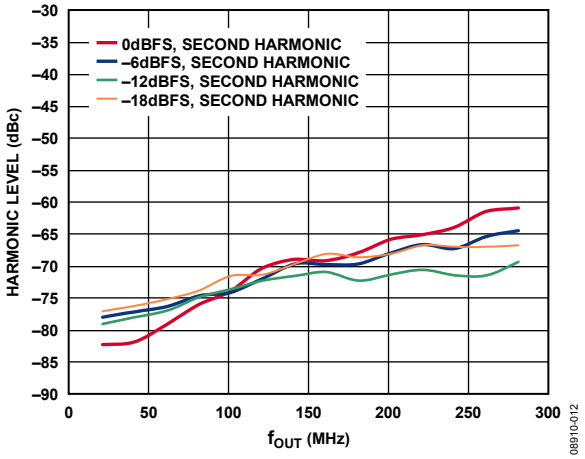



Figure 12. Second Harmonic vs. $f_{OUT}$ over Digital Scale, Full-Scale Current = 20 mA, 4× Interpolation, $f_{DATA}$ = 150 MSPS

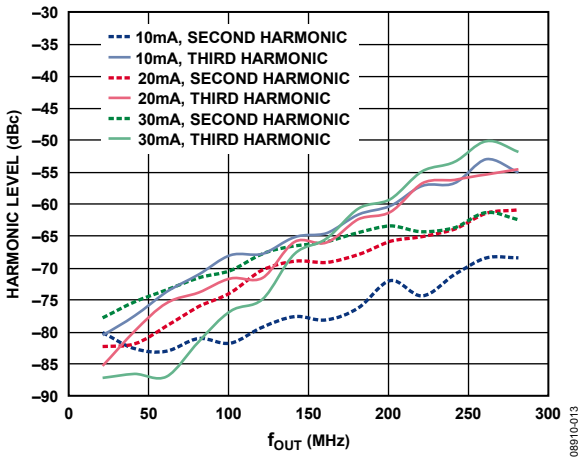



Figure 13. Second Harmonic vs. $f_{OUT}$ over Full-Scale Current, Digital Scale = 0 dBFS, 4× Interpolation, $f_{DATA}$ = 150 MSPS

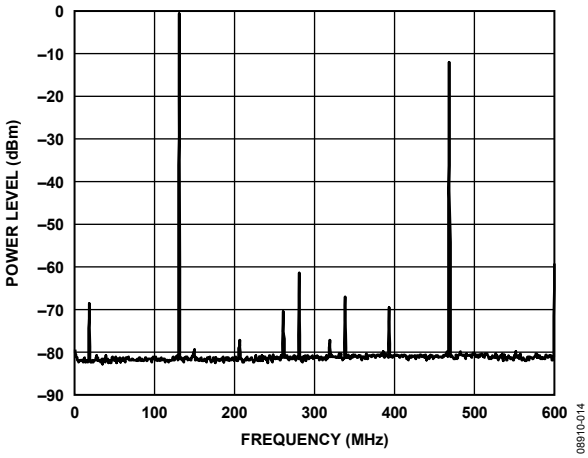



Figure 14. 4× Interpolation, $f_{DATA}$ = 150 MSPS, $f_{OUT}$ = 131 MHz

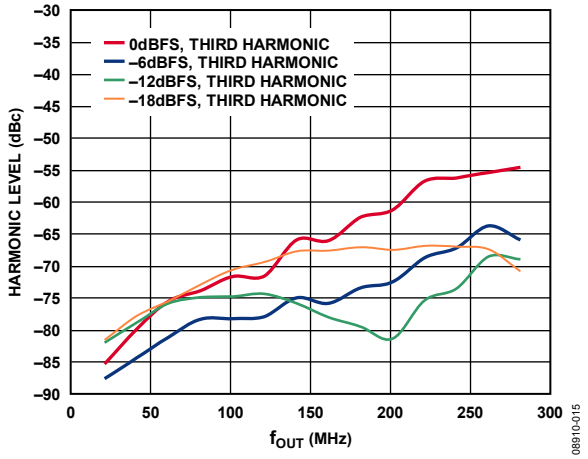



Figure 15. Third Harmonic vs. $f_{OUT}$ over Digital Scale, Full-Scale Current = 20 mA, 4× Interpolation, $f_{DATA}$ = 150 MSPS

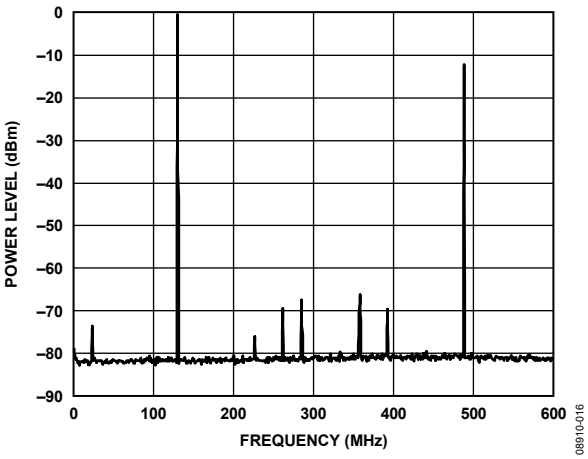



Figure 16. 2× Interpolation, $f_{DATA}$ = 310 MSPS, $f_{OUT}$ = 131 MHz

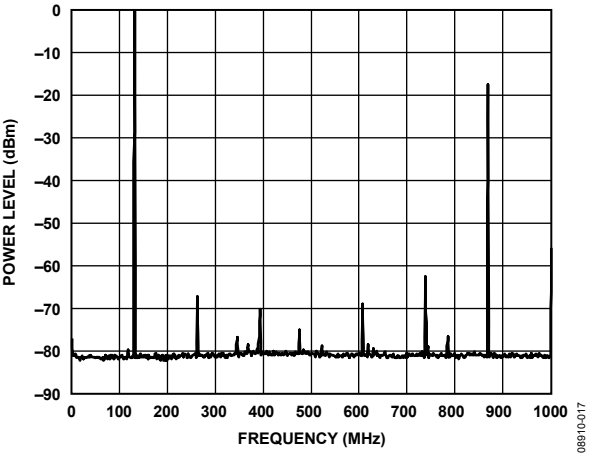



Figure 17. 8× Interpolation, $f_{DATA}$ = 125 MSPS, $f_{OUT}$ = 131 MHz

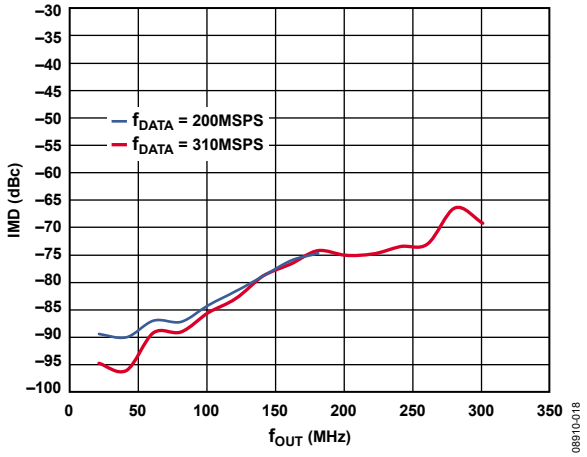



Figure 18. IMD vs. $f_{OUT}$ over $f_{DATA}$, 2× Interpolation, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

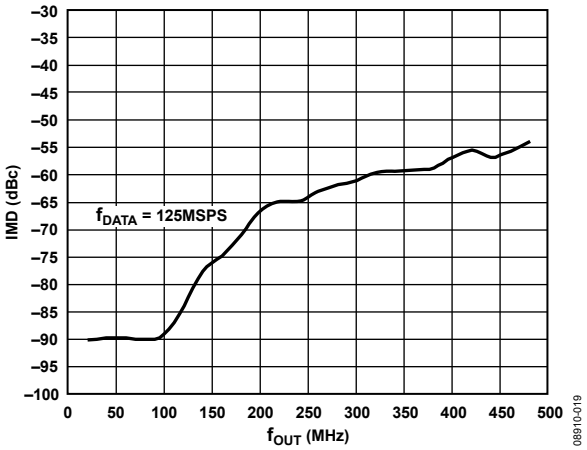



Figure 19. IMD vs. $f_{OUT}$, 8× Interpolation, $f_{DATA}$ = 125 MSPS, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

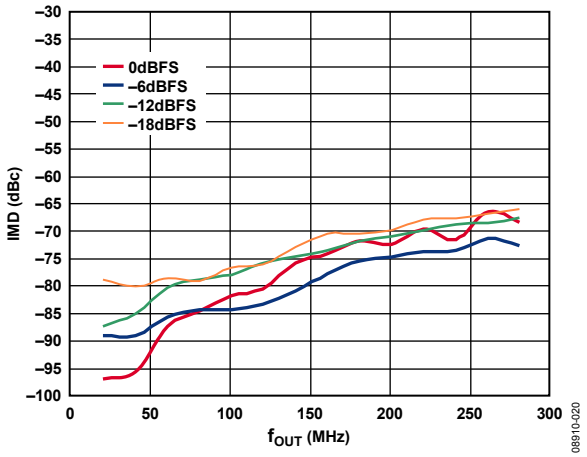



Figure 20. IMD vs. $f_{OUT}$ over Digital Scale, 4× Interpolation, $f_{DATA}$ = 150 MSPS, Full-Scale Current = 20 mA

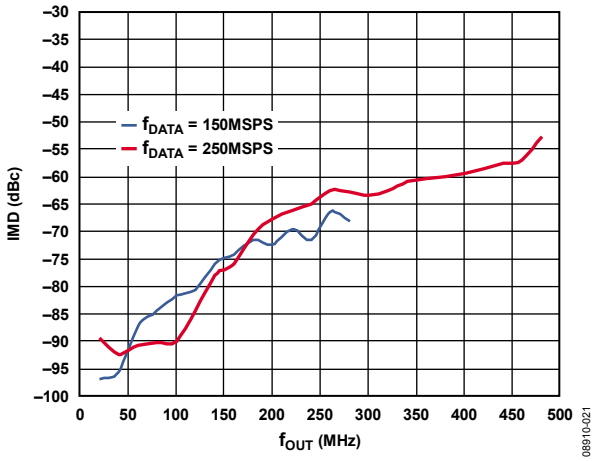



Figure 21. IMD vs. $f_{OUT}$ over $f_{DATA}$, 4× Interpolation, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

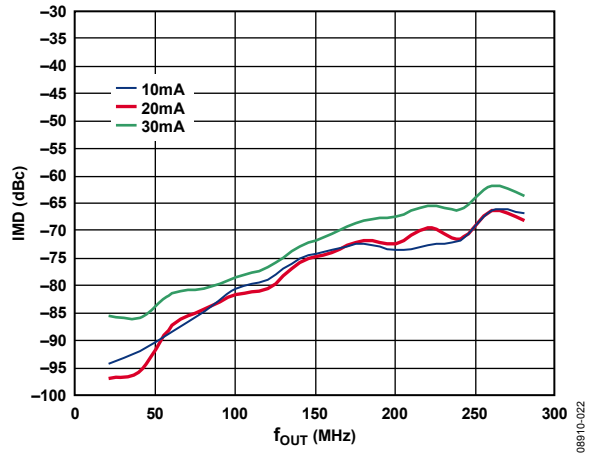



Figure 22. IMD vs. $f_{OUT}$ over Full-Scale Current, 4× Interpolation, $f_{DATA}$ = 150 MSPS, Digital Scale = 0 dBFS

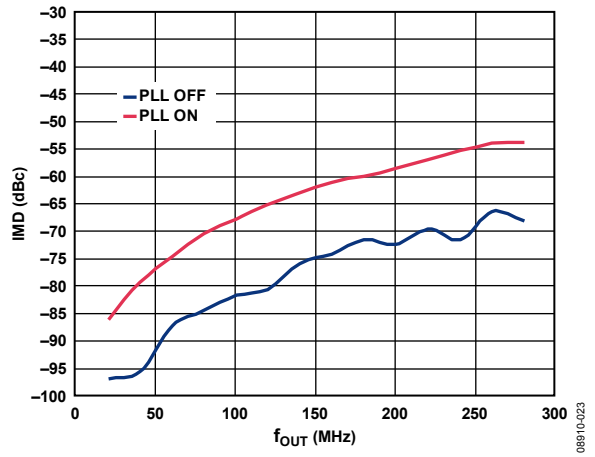



Figure 23. IMD vs. $f_{OUT}$, PLL On and Off, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

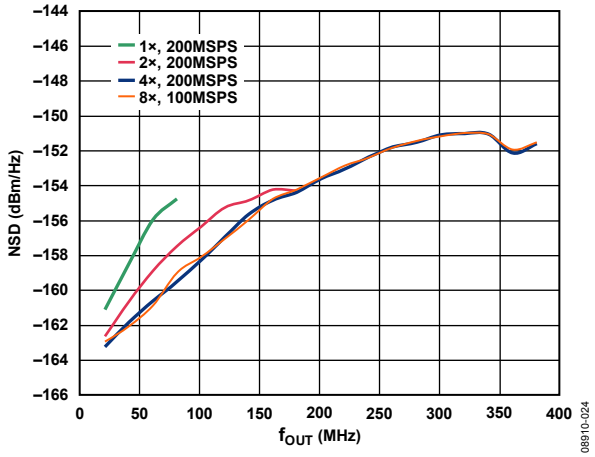



Figure 24. Single-Tone NSD Performance vs. $f_{OUT}$, Digital Scale = 0 dBFS, 4× $f_{DATA}$ = 200 MSPS, Full-Scale Current = 20 mA

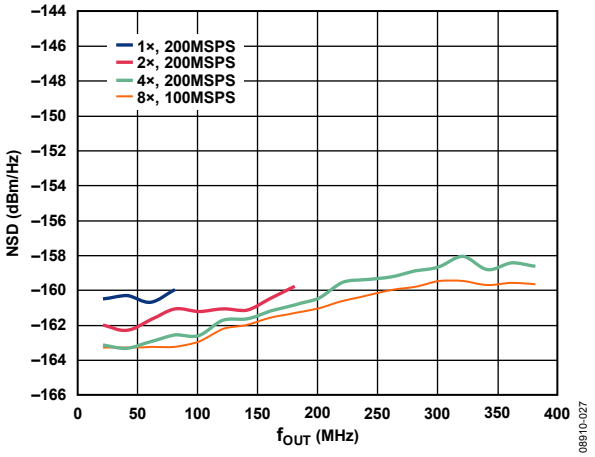



Figure 27. Eight-Tone NSD Performance vs. $f_{OUT}$, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA

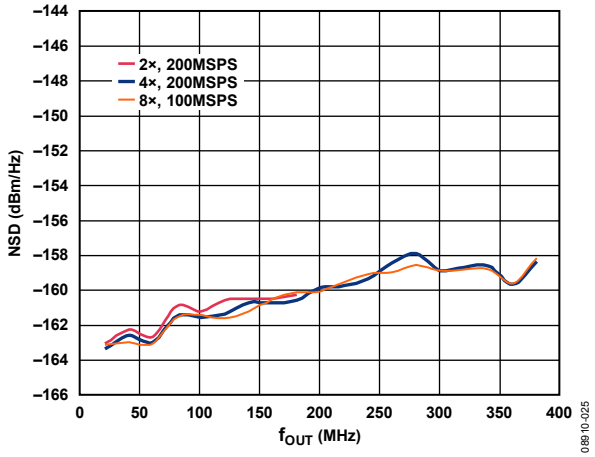



Figure 25. Single-Tone NSD Performance vs. $f_{OUT}$, Digital Scale = 0 dBFS, 4× $f_{DATA}$ = 200 MSPS, Full-Scale Current = 20 mA, PLL On

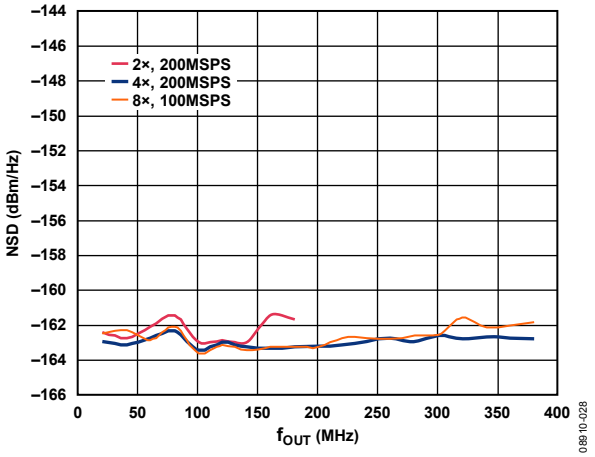



Figure 28. Single-Tone NSD Performance vs. $f_{OUT}$, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA, PLL On

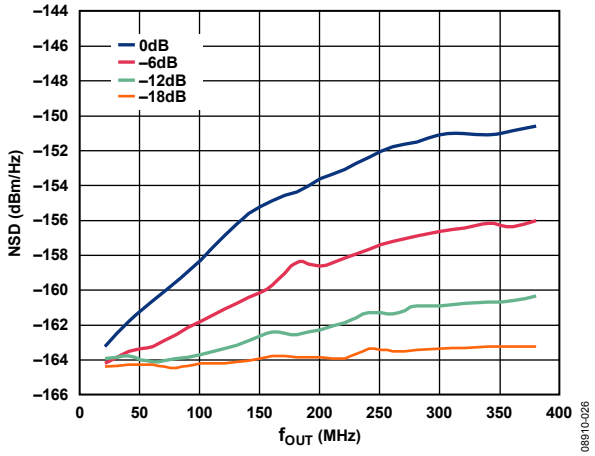



Figure 26. Single-Tone NSD Performance vs. $f_{OUT}$ over Digital Scale, 4× $f_{DATA}$ = 200 MSPS, Full-Scale Current = 20 mA

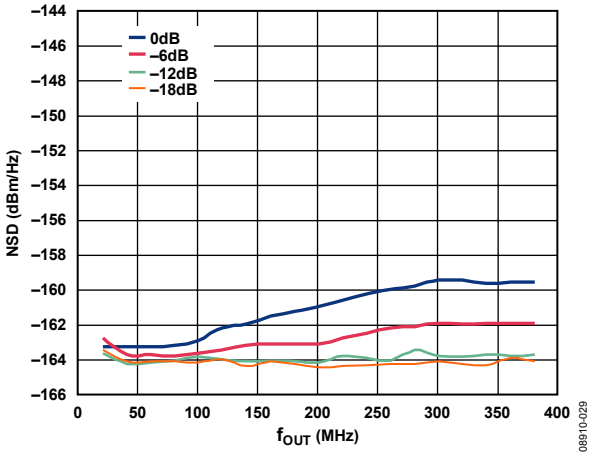



Figure 29. Eight-Tone NSD Performance vs. $f_{OUT}$ over Digital Scale, 4× $f_{DATA}$ = 200 MSPS, Full-Scale Current = 20 mA

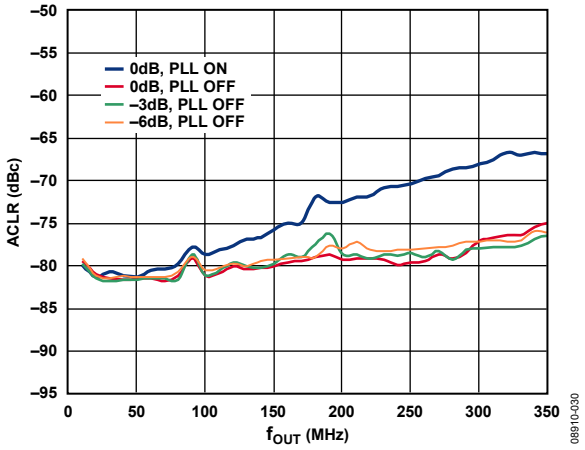



*Figure 30. One-Carrier W-CDMA ACLR vs. $f_{OUT}$, Adjacent Channel, 4× Interpolation, $f_{DATA}$ = 184.32 MHz*

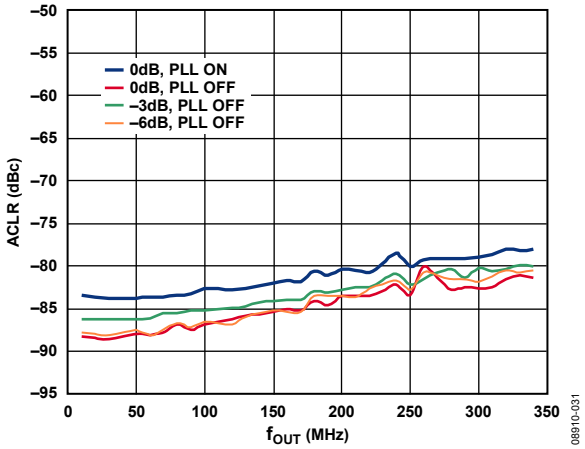



*Figure 31. One-Carrier W-CDMA ACLR vs. $f_{OUT}$, Alternate Channel, 4× Interpolation, $f_{DATA}$ = 184.32 MHz*

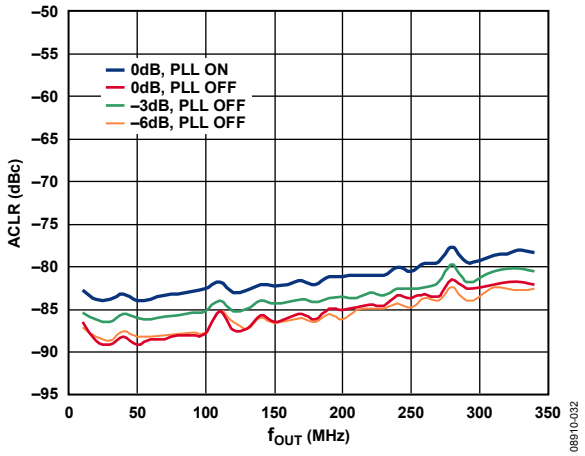



*Figure 32. One-Carrier W-CDMA ACLR vs. $f_{OUT}$, Second Alternate Channel, 4× Interpolation, $f_{DATA}$ = 184.32 MHz*

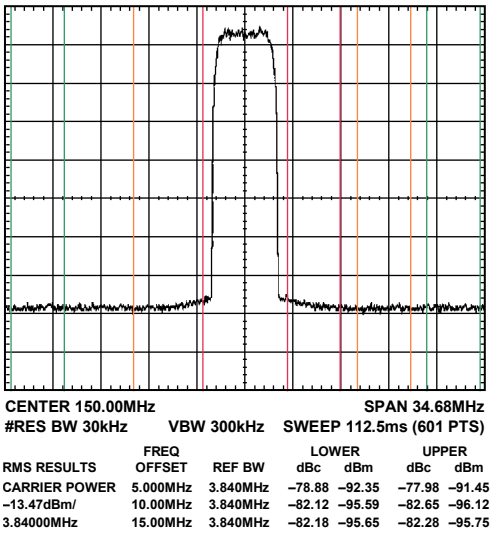

CENTER 150.00MHz
SPAN 34.68MHz
#RES BW 30kHz
VBW 300kHz
SWEEP 112.5ms (601 PTS)
FREQ LOWER UPPER
RMS RESULTS OFFSET REF BW dBc dBm dBc dBm
CARRIER POWER 5.000MHz 3.840MHz –78.88 –92.35 –77.98 –91.45
–13.47dBm/ 10.00MHz 3.840MHz –82.12 –95.59 –82.65 –96.12
3.84000MHz 15.00MHz 3.840MHz –82.18 –95.65 –82.28 –95.75
08910-033

*Figure 33. One-Carrier W-CDMA ACLR, $f_{OUT}$ = 150 MHz, 4× Interpolation, $f_{DATA}$ = 184.32 MHz, PLL Off*

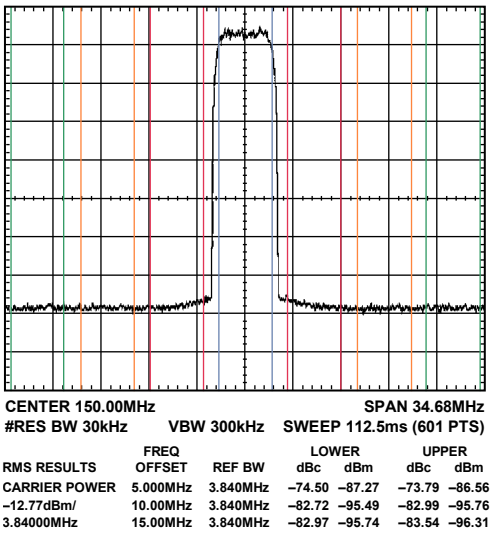

CENTER 150.00MHz
SPAN 34.68MHz
#RES BW 30kHz
VBW 300kHz
SWEEP 112.5ms (601 PTS)
FREQ LOWER UPPER
RMS RESULTS OFFSET REF BW dBc dBm dBc dBm
CARRIER POWER 5.000MHz 3.840MHz –74.50 –87.27 –73.79 –86.56
–12.77dBm/ 10.00MHz 3.840MHz –82.72 –95.49 –82.99 –95.76
3.84000MHz 15.00MHz 3.840MHz –82.97 –95.74 –83.54 –96.31
08910-034

*Figure 34. One-Carrier W-CDMA ACLR, $f_{OUT}$ = 150 MHz, 4× Interpolation, $f_{DATA}$ = 184.32 MHz, PLL On*

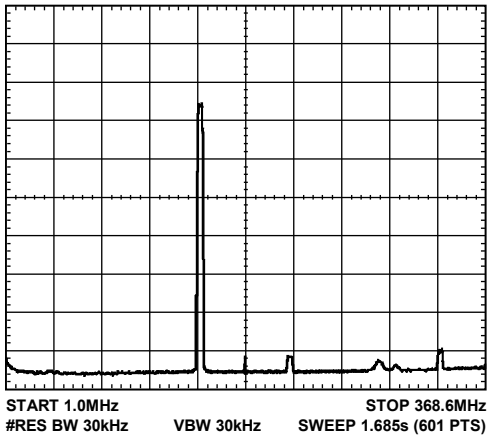



*Figure 35. One-Carrier W-CDMA, $f_{OUT}$ = 150 MHz, $f_{DAC}$ = 737.28 MSPS, 4× Interpolation, –3 dBFS*

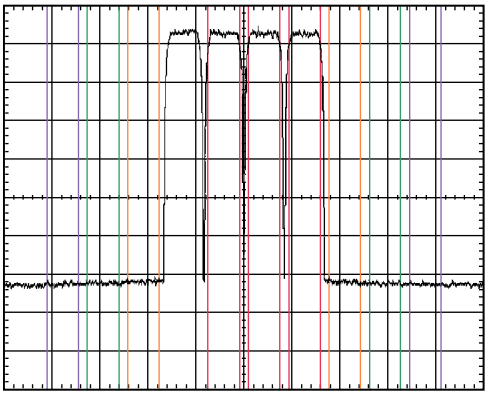



*Figure 36. Four-Carrier W-CDMA, $f_{OUT}$ = 150 MHz, $f_{DAC}$ = 737.28 MSPS, 4× Interpolation, −3 dBFS, PLL Off*

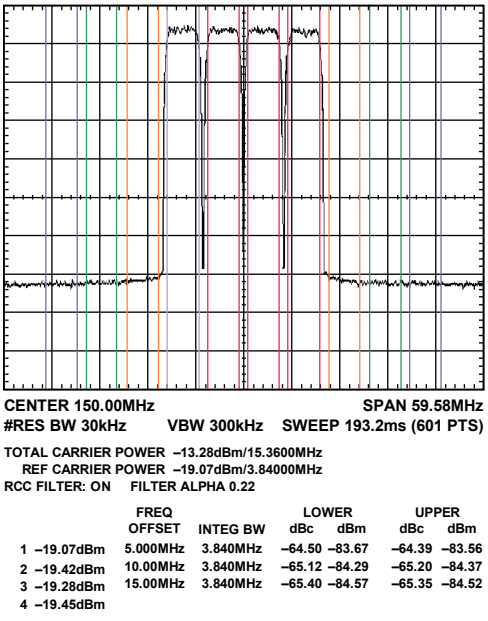



*Figure 37. Four-Carrier W-CDMA, $f_{OUT}$ = 150 MHz, $f_{DAC}$ = 737.28 MSPS, 4× Interpolation, −3 dBFS, PLL On*

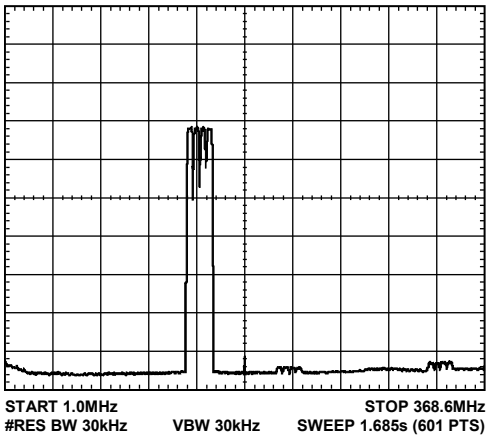



*Figure 38. Four-Carrier W-CDMA, $f_{OUT}$ = 150 MHz, $f_{DAC}$ = 737.28 MSPS, 4× Interpolation, −3 dBFS*

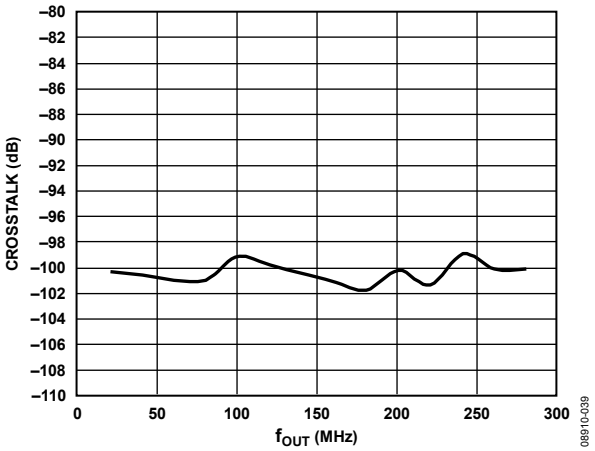



*Figure 39. Crosstalk (DAC Set 1 to DAC Set 2), 4× Interpolation, $f_{DATA}$ = 150 MSPS, Digital Scale = 0 dBFS, Full-Scale Current = 20 mA*

# TERMINOLOGY

**Integral Nonlinearity (INL)**
INL is defined as the maximum deviation of the actual analog output from the ideal output, determined by a straight line drawn from zero scale to full scale.

**Differential Nonlinearity (DNL)**
DNL is the measure of the variation in analog value, normalized to full scale, associated with a 1 LSB change in digital input code.

**Monotonicity**
A DAC is monotonic if the output either increases or remains constant as the digital input increases.

**Offset Error**
The deviation of the output current from the ideal of zero is called offset error. For IOUTx_P, 0 mA output is expected when the inputs are all 0s. For IOUTx_N, 0 mA output is expected when all inputs are set to 1.

**Gain Error**
The difference between the actual and ideal output span. The actual span is determined by the difference between the output when all inputs are set to 1 and the output when all inputs are set to 0.

**Output Compliance Range**
The range of allowable voltage at the output of a current-output DAC. Operation beyond the maximum compliance limits can cause either output stage saturation or breakdown, resulting in nonlinear performance.

**Temperature Drift**
Temperature drift is specified as the maximum change from the ambient (25°C) value to the value at either $T_{MIN}$ or $T_{MAX}$. For offset and gain drift, the drift is reported in ppm of full-scale range (FSR) per degrees Celsius. For reference drift, the drift is reported in ppm per degrees Celsius.

**Power Supply Rejection (PSR)**
The maximum change in the full-scale output as the supplies are varied from minimum to maximum specified voltages.

**Settling Time**
The time required for the output to reach and remain within a specified error band around its final value, measured from the start of the output transition.

**In-Band Spurious Free Dynamic Range (SFDR)**
The difference, in decibels, between the peak amplitude of the output signal and the peak spurious signal between dc and the frequency equal to half the input data rate.

**Out-of-Band Spurious Free Dynamic Range (SFDR)**
The difference, in decibels, between the peak amplitude of the output signal and the peak spurious signal within the band that starts at the frequency of the input data rate and ends at the Nyquist frequency of the DAC output sample rate. Normally, energy in this band is rejected by the interpolation filters. This specification, therefore, defines how well the interpolation filters work and the effect of other parasitic coupling paths on the DAC output.

**Total Harmonic Distortion (THD)**
THD is the ratio of the rms sum of the first six harmonic components to the rms value of the measured fundamental. It is expressed as a percentage or in decibels.

**Signal-to-Noise Ratio (SNR)**
SNR is the ratio of the rms value of the measured output signal to the rms sum of all other spectral components below the Nyquist frequency, excluding the first six harmonics and dc. The value for SNR is expressed in decibels.

**Interpolation Filter**
An interpolation filter up-samples the input digital data by a multiple of $f_{DATA}$ (interpolation rate) and then filters out the undesired spectral images created by the up-sampling process.

**Adjacent Channel Leakage Ratio (ACLR)**
The ratio in dBc between the measured power within a channel relative to its adjacent channel.

**Complex Image Rejection**
In a traditional two-part upconversion, two images are created around the second IF frequency. These images have the effect of wasting transmitter power and system bandwidth. By placing the real part of a second complex modulator in series with the first complex modulator, either the upper or lower frequency image near the second IF can be rejected.

# SERIAL PERIPHERAL INTERFACE

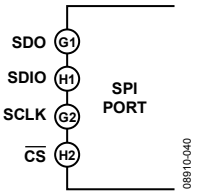



*Figure 40. SPI Port*

The serial port is a flexible, synchronous serial communications port allowing easy interface to many industry-standard microcontrollers and microprocessors. The serial I/O is compatible with most synchronous transfer formats, including both the Motorola SPI and Intel® SSR protocols. The interface allows read/write access to all registers that configure the AD9148. Single- or multiple-byte transfers are supported, as well as MSB-first or LSB-first transfer formats. The serial interface ports can be configured as a single pin I/O (SDIO) or two unidirectional pins for input/output (SDIO/SDO).

## GENERAL OPERATION OF THE SERIAL INTERFACE

There are two phases to a communication cycle with the AD9148. Phase 1 is the instruction cycle (the writing of an instruction byte into the device), coincident with the first eight SCLK rising edges. The instruction byte provides the serial port controller with information regarding the data transfer cycle, Phase 2 of the communication cycle. The Phase 1 instruction byte defines whether the upcoming data transfer is a read or a write, and the starting register address for the first byte of the data transfer. The first eight SCLK rising edges of each communication cycle are used to write the instruction byte into the device.

A logic high on the $\overline{CS}$ pin followed by a logic low resets the SPI port timing to the initial state of the instruction cycle. From this state, the next eight rising SCLK edges represent the instruction bits of the current I/O operation, regardless of the state of the internal registers or the other signal levels at the inputs to the SPI port. If the SPI port is in an instruction cycle or a data transfer cycle, none of the present data is written.

The remaining SCLK edges are for Phase 2 of the communication cycle. Phase 2 is the actual data transfer between the device and the system controller. Phase 2 of the communication cycle is a transfer of one or more data bytes. Registers change immediately upon writing to the last bit of each transfer byte.

## DATA FORMAT

The instruction byte contains the information shown in Table 11.

**Table 11. SPI Instruction Byte**

| I7 (MSB) | I6 | I5 | I4 | I3 | I2 | I1 | I0 (LSB) |
|---|---|---|---|---|---|---|---|
| R/$\overline{W}$ | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

R/$\overline{W}$, Bit 7 of the instruction byte, determines whether a read or a write data transfer occurs after the instruction byte write. Logic high indicates a read operation, and Logic 0 indicates a write operation.

A6 through A0, Bit 6 through Bit 0 of the instruction byte, determine the register that is accessed during the data transfer portion of the communication cycle. For multibyte transfers, this address is the starting byte address. The remaining register addresses are generated by the device based on the LSB-first bit (Register 0x00, Bit 6).

## SPI PIN DESCRIPTIONS

### *Serial Clock (SCLK)*

The serial clock pin synchronizes data to and from the device and runs the internal state machines. The maximum frequency of SCLK is 40 MHz. All data input is registered on the rising edge of SCLK. All data is driven out on the falling edge of SCLK.

### *Chip Select ($\overline{CS}$)*

Active low input starts and gates a communication cycle. It allows more than one device to be used on the same serial communications lines. The SDO and SDIO pins go to a high impedance state when this input is high. Chip select should stay low during the entire communication cycle.

### *Serial Data I/O (SDIO)*

Data is always written into the device on this pin. However, this pin can be used as a bidirectional data line. The configuration of this pin is controlled by Register 0x00, Bit 7. The default is Logic 0, configuring the SDIO pin as unidirectional.

### *Serial Data Output (SDO)*

Data is read from this pin for protocols that use separate lines for transmitting and receiving data. In the case where the device operates in a single bidirectional I/O mode, this pin does not output data and is set to a high impedance state.

## SPI OPTIONS

The serial port can support both MSB-first and LSB-first data formats. This functionality is controlled by the LSB first bit (Register 0x00, Bit 6). The default is MSB first (LSB first = 0).

When LSB first = 0 (MSB first), the instruction and data bit must be written from MSB to LSB. Multibyte data transfers in MSB-first format start with an instruction byte that includes the register address of the most significant data byte. Subsequent data bytes should follow from the high address to the low address. In MSB-first mode, the serial port internal byte address generator decrements for each data byte of the multibyte communication cycle.

When LSB first = 1 (LSB first), the instruction and data bit must be written from LSB to MSB. Multibyte data transfers in LSB-first format start with an instruction byte that includes the register address of the least significant data byte followed by multiple data bytes. The serial port internal byte address generator increments for each byte of the multibyte communication cycle.

The serial port controller data address decrements from the data address written toward 0x00 for multibyte I/O operations if the MSB-first mode is active. The serial port controller address increments from the data address written toward 0x1F for multibyte I/O operations if the LSB-first mode is active.

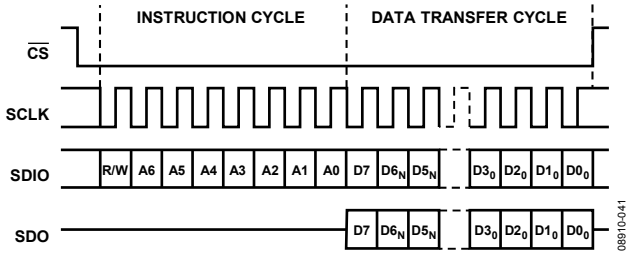



*Figure 41. Serial Register Interface Timing MSB First*

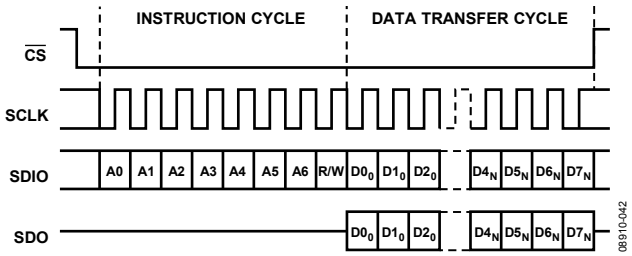



*Figure 42. Serial Register Interface Timing LSB First*

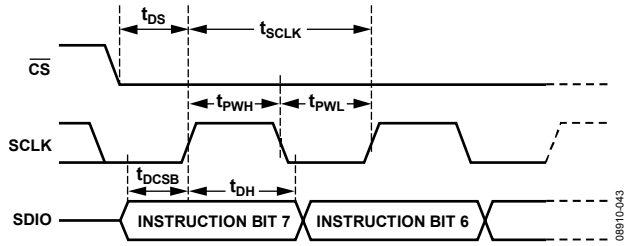



*Figure 43. Timing Diagram for SPI Register Write*

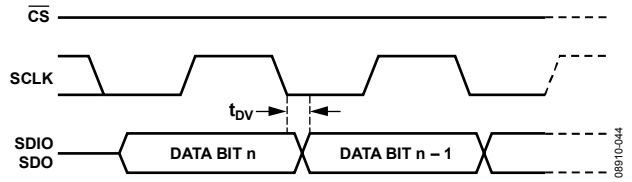



*Figure 44. Timing Diagram for SPI Register Read*

# SPI REGISTER MAP

**Table 12. Register Map**

| Addr | Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | Comm | SDIO direction | LSB/ MSB first | Software reset | DAC SPI select | | | | | 0x00 |
| 0x01 | Power control | Power-Down DAC Set 1 | Power-Down DAC Set 2 | Power-down data receiver | | | | | | 0x00 |
| 0x03 | Data format | Binary format | Q first enable | Dual-port mode | Bus swap | Byte mode | Byte swap | | | 0x20 |
| 0x04 | Interrupt Enable 0 | Enable PLL lock lost | Enable PLL lock | Enable sync lock lost | Enable sync lock | | Enable FIFO SPI aligned | Enable FIFO Warning 1 | Enable FIFO Warning 2 | 0x00 |
| 0x05 | Interrupt Enable 1 | | | | Enable AED compare pass | Enable AED compare fail | Enable SED compare fail | | | 0x00 |
| 0x06 | Event Flag 0 | PLL lock lost | PLL lock | Sync lock lost | Sync lock | | FIFO SPI aligned | FIFO Warning 1 | FIFO Warning 2 | |
| 0x07 | Event Flag 1 | | | | AED compare pass | AED compare fail | SED compare fail | | | |
| 0x08 | Clock receiver control | CLK duty correction | REFCLK duty correction | CLK cross correction | REFCLK cross correction | 0 | 1 | 1 | 1 | 0x37 |
| 0x0A | PLL Control 0 | PLL enable | PLL manual enable | Manual VCO Band[5:0] | | | | | | 0x40 |
| 0x0C | PLL Control 1 | PLL Loop Bandwidth[2:0] | | | 0 | 1 | 0 | 0 | 1 | 0xF1 |
| 0x0D | PLL Control 2 | N2[1:0] | | | PLL cross control enable | N0[1:0] | | N1[1:0] | | 0xD9 |
| 0x0E | PLL Status 0 | | | | | PLL Control Voltage[3:0] | | | | |
| 0x0F | PLL Status 1 | | | VCO Band Readback[5:0] | | | | | | |
| 0x10 | Sync Control 0 | Sync enable | FIFO rate/ data rate toggle | | | Rising edge sync | Sync Averaging[2:0] | | | 0x08 |
| 0x11 | Sync Control 1 | | | Sync Phase Request[5:0] | | | | | | 0x00 |
| 0x12 | Sync Status 0 | Sync lost | Sync locked | | | | | | | |
| 0x14 | Data receiver control | | One DCI | | | | | | | 0x00 |
| 0x15 | Data receiver status | LVDS rcvr frame high | LVDS rcvr frame low | LVDS rcvr DCI high | LVDS rcvr DCI low | LVDS rcvr Port B high | LVDS rcvr Port B low | LVDS rcvr Port A high | LVDS rcvr Port A low | |
| 0x17 | FIFO Status/ Control Port A | FIFO Warning 1 | FIFO Warning 2 | FIFO reset aligned | FIFO SPI align ack | FIFO SPI align requesting | FIFO Phase Offset[2:0] | | | 0x00 |
| 0x18 | FIFO Status Port A | FIFO Level[7:0] | | | | | | | | |
| 0x19 | FIFO Status/ Control Port B | FIFO Warning 1 | FIFO Warning 2 | FIFO reset aligned | FIFO SPI align ack | FIFO SPI align requesting | FIFO Phase Offset[2:0] | | | 0x00 |
| 0x1A | FIFO Status Port B | FIFO Level[7:0] | | | | | | | | |
| 0x1C | HB1 control | Enable pre mod | Bypass $sinc^{-1}$ | | | | HB1[1:0] | | Bypass HB1 | 0x40 |
| 0x1D | HB2 control | | | | | HB2[2:0] | | | Bypass HB2 | 0x00 |
| 0x1E | HB3 control | Bypass digital gain and phase adjustment | | | | HB3[2:0] | | | Bypass HB3 | 0x81 |

| Addr | Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x1F | Chip ID | Chip ID[7:0] | | | | | | | | 0x20 |
| 0x20[1] | Coeff I Byte 0 | 0 | Coeff_1i[3:0] | | | | Coeff_0i[2:0] | | | 0x00 |
| 0x21[1] | Coeff I Byte 1 | Coeff_3i[2:0] | | | Coeff_2i[4:0] | | | | | 0xC0 |
| 0x22[1] | Coeff I Byte 2 | Coeff_4i[2:0] | | | 0 | Coeff_3i[6:3] | | | | 0xEF |
| 0x23[1] | Coeff I Byte 3 | 0 | Coeff_4i[9:3] | | | | | | | 0x7F |
| 0x24[1] | Coeff Q Byte 0 | 0 | Coeff_1q[3:0] | | | | Coeff_0q[2:0] | | | 0x69 |
| 0x25[1] | Coeff Q Byte 1 | Coeff_3q[2:0] | | | Coeff_2q[4:0] | | | | | 0xE6 |
| 0x26[1] | Coeff Q Byte 2 | Coeff_4q[2:0] | | | 0 | Coeff_3q[6:3] | | | | 0x0D |
| 0x27[1] | Coeff Q Byte 3 | 0 | Coeff_4q[9:3] | | | | | | | 0x00 |
| 0x28[1] | I phase adj LSB | Phase Word I[7:0] | | | | | | | | 0x00 |
| 0x29[1] | I phase adj MSB | | | | | | | Phase Word I[9:8] | | 0x00 |
| 0x2A[1] | Q phase adj LSB | Phase Word Q[7:0] | | | | | | | | 0x00 |
| 0x2B[1] | Q phase adj MSB | | | | | | | Phase Word Q[9:8] | | 0x00 |
| 0x2C[1] | I DC offset LSB | DC Offset I[7:0] | | | | | | | | 0x00 |
| 0x2D[1] | I DC offset MSB | DC Offset I[15:8] | | | | | | | | 0x00 |
| 0x2E[1] | Q DC offset LSB | DC Offset Q[7:0] | | | | | | | | 0x00 |
| 0x2F[1] | Q DC offset MSB | DC Offset Q[15:8] | | | | | | | | 0x00 |
| 0x30[1] | IDAC FSC adj | IDAC FSC Adj[7:0] | | | | | | | | 0xF9 |
| 0x31[1] | IDAC control | IDAC sleep | | | | | | IDAC FSC Adj[9:8] | | 0x01 |
| 0x32[1] | AUX IDAC data | AUX IDAC Data[7:0] | | | | | | | | 0x00 |
| 0x33[1] | AUX IDAC control | AUX IDAC sign | AUX IDAC current direction | AUX IDAC power-down | | | | AUX IDAC Data[9:8] | | 0x00 |
| 0x34[1] | QDAC FSC adj | QDAC FSC Adj[7:0] | | | | | | | | 0xF9 |
| 0x35[1] | QDAC control | QDAC sleep | | | | | | QDAC FSC Adj[9:8] | | 0x01 |
| 0x36[1] | AUX QDAC data | AUX QDAC Data[7:0] | | | | | | | | 0x00 |
| 0x37[1] | AUX QDAC control | AUX QDAC sign | AUX QDAC current direction | AUX QDAC power-down | | | | AUX QDAC Data[9:8] | | 0x00 |
| 0x38[1] | SED_S0_L | SED Compare Pattern Sample0[7:0] | | | | | | | | 0xB6 |
| 0x39[1] | SED_S0_H | SED Compare Pattern Sample0[15:8] | | | | | | | | 0x7A |
| 0x3A[1] | SED_S1_L | SED Compare Pattern Sample1[7:0] | | | | | | | | 0x45 |
| 0x3B[1] | SED_S1_H | SED Compare Pattern Sample1[15:8] | | | | | | | | 0xEA |
| 0x3C[1] | SED3_S2_L | SED Compare Pattern Sample2[7:0] | | | | | | | | 0x16 |
| 0x3D[1] | SED3_S2_H | SED Compare Pattern Sample2[15:8] | | | | | | | | 0x1A |
| 0x3E[1] | SED4_S3_L | SED Compare Pattern Sample3[7:0] | | | | | | | | 0xC6 |
| 0x3F[1] | SED4_S3_H | SED Compare Pattern Sample3[15:8] | | | | | | | | 0xAA |
| 0x40 | SED control/ status | SED compare enable | Port B error detected | Port A error detected | | Auto-clear enable | Port B compare failed | Port A compare failed | Compare passed | 0x00 |
| 0x41[1] | SED_R_L | SED Status Rising Edge Samples[7:0] | | | | | | | | |
| 0x42[1] | SED_R_H | SED Status Rising Edge Samples[15:8] | | | | | | | | |
| 0x43[1] | SED_F_L | SED Status Falling Edge Samples[7:0] | | | | | | | | |
| 0x44[1] | SED_F_H | SED Status Falling Edge Samples[15:8] | | | | | | | | |
| 0x50[1] | I gain control | I Gain[7:0] | | | | | | | | 0x40 |
| 0x51[1] | Q gain control | Q Gain[7:0] | | | | | | | | 0x40 |

| Addr | Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x54 | FTW (LSB) | FTW[7:0] | | | | | | | | 0x00 |
| 0x55 | FTW | FTW [15:8] | | | | | | | | 0x00 |
| 0x56 | FTW | FTW[23:16] | | | | | | | | 0x00 |
| 0x57 | FTW (MSB) | FTW[31:24] | | | | | | | | 0x00 |
| 0x58 | Phase offset (MSB) | NCO Phase Offset[15:8] | | | | | | | | 0x00 |
| 0x59 | Phase offset (LSB) | NCO Phase Offset[7:0] | | | | | | | | 0x00 |
| 0x5A | DDS/mod control | Bypass DDS/MOD | | Frame NCO reset ack | Frame NCO reset request | FTW update ack | FTW update request | | Sideband select | 0x80 |
| 0x5C | Die Temp Control 0 | | | | | | | Latch temp data | Temp sensor power-down | 0x01 |
| 0x5D | Die Temp Control 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0x20 |
| 0x5E | Die temp LSB | Die Temp[7:0] | | | | | | | | |
| 0x5F | Die temp MSB | Die Temp[15:8] | | | | | | | | |
| 0x72 | DCI delay | | | | | | | DCI Delay[1:0] | | 0x00 |

[1] Register 0x20 to Register 0x3F and Register 0x41 to Register 0x51 configure DAC 1 (I) and DAC 2 (Q) data paths with DAC SPI select = 0 (Register 0x00[4]). Register 0x20 to Register 0x3F and Register 0x41 to Register 0x51 configure DAC 3 (I) and DAC 4 (Q) data paths with DAC SPI select = 1 (Register 0x00[4]).

## SPI REGISTER DESCRIPTIONS

**Table 13. Register Descriptions**

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| Comm | 00 | 7 | SDIO | SDIO operation.<br>0 = SDIO operates as an input only.<br>1 = SDIO operates as bidirectional input/output. | 0 |
| | | 6 | LSB/MSB first | SPI communication LSB first (default is MSB first).<br>0 = MSB first.<br>1 = LSB first. | 0 |
| | | 5 | Software Reset | Software reset.<br>Reset is asserted when this bit transitions from 0 to 1. | 0 |
| | | 4 | DAC SPI select | Selects which DAC data path Register 0x20 to Register 0x3F and Register 0x41 to Register 0x51 configure.<br>0 = DAC 1 (I path) and DAC 2 (Q path) are configured.<br>1 = DAC 3 (I path) and DAC 4 (Q path) are configured. | 0<br>0 |
| Power Control | 01 | 7 | Power-Down DAC Set 1 | Power down DAC 1 and power down DAC 2. | 0 |
| | | 6 | Power-Down DAC Set 2 | Power down DAC 3 and power down DAC 4. | 0 |
| | | 5 | Power-down data receiver | Power down the input data receiver. | 0 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| Data Format | 03 | 7 | Binary format | Input data is in twos complement format (0) or unsigned binary format (1). | 0 |
| | | 6 | Q first enable | Indicates I/Q data pairing on data input; I first (0), Q first (1). | 0 |
| | | 5 | Dual-port mode | Number of input data ports used.<br>Single port (0), dual port (1). | 1 |
| | | 4 | Bus swap | 0 = normal data input bus pin out (MSB to LSB).<br>1 = inverted data input bus pin out (LSB to MSB). | 0 |
| | | 3 | Byte mode | 0 = data input bus is 16-bit wide on each port.<br>1 = data input bus is two 8-bit wide buses on Port A. | 0 |
| | | 2 | Byte swap | 0 = normal data input bus pin out (MSB to LSB).<br>1 = inverted data input bus pin out (LSB to MSB). | 0 |
| Interrupt Enable 0 | 04 | 7 | Enable PLL lock lost | Enables interrupt for PLL lock lost. | 0 |
| | | 6 | Enable PLL lock | Enables interrupt for PLL lock. | 0 |
| | | 5 | Enable sync lock lost | Enables interrupt for sync lock lost. | 0 |
| | | 4 | Enable sync lock | Enables interrupt for sync lock. | 0 |
| | | 2 | Enable FIFO SPI aligned | Enables interrupt for FIFO SPI aligned. | 0 |
| | | 1 | Enable FIFO Warning 1 | Enables interrupt for FIFO Warning 1. | 0 |
| | | 0 | Enable FIFO Warning 2 | Enables interrupt for FIFO Warning 2. | 0 |
| Interrupt Enable 1 | 05 | 4 | Enable AED compare pass | Enables interrupt for AED compare pass. | 0 |
| | | 3 | Enable AED compare fail | Enables interrupt for AED compare fail. | 0 |
| | | 2 | Enable SED compare fail | Enables interrupt for SED compare fail. | 0 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| Event Flag 0 (All bits are high when interrupt is active. Clear interrupt by writing respective bit high.) | 06 | 7 | PLL lock lost | 1 = indicates that the PLL that was previously locked has unlocked from the reference signal. | 0 |
| | | 6 | PLL lock | 1 = indicates that the PLL has locked to the reference clock input. | 0 |
| | | 5 | Sync lock lost | 1 = indicates that the sync logic that was previously locked has lost alignment. | 0 |
| | | 4 | Sync lock | 1 = indicates that the sync logic achieved sync alignment. This is indicated when no phase changes are requested for at least a few full averaging cycles. | 0 |
| | | 2 | FIFO SPI aligned | 1 = indicates that a FIFO reset originating from a serial port-based request has successfully completed. | 0 |
| | | 1 | FIFO Warning 1 | 1 = indicates that the difference between the FIFO read and write pointers is 1. | 0 |
| | | 0 | FIFO Warning 2 | 1 = indicates that the difference between the FIFO read and write pointers is 2. | 0 |
| Event Flag 1(All bits are high when interrupt is active. Clear interrupt by writing respective bit high). | 07 | 4 | AED compare pass | 1 = indicates that the SED logic detected a valid input data pattern comparison against the preprogrammed expected values. | 0 |
| | | 3 | AED compare fail | 1 = indicates that the SED logic detected an invalid input data pattern comparison against the preprogrammed expected values. This automatically clears when eight valid I/Q data pairs are received. | 0 |
| | | 2 | SED compare fail | 1 = indicates that the SED logic detected an invalid input data pattern comparison against the preprogrammed expected values. | 0 |
| Clock Receiver Control | 08 | 7 | CLK duty correction | Enables duty-cycle correction on CLK input. | 0 |
| | | 6 | REFCLK duty correction | Enables duty-cycle correction on REFCLK input. | 0 |
| | | 5 | CLK cross correction | Enables differential crossing correction on CLK input. | 1 |
| | | 4 | REFCLK cross correction | Enables differential crossing correction on REFCLK input. | 1 |
| | | 3:0 | 0111 | Always set these bits to 0111 | 0111 |
| PLL Control 0 | 0A | 7 | PLL enable | Enables PLL clock multiplier. | 0 |
| | | 6 | PLL manual enable | Enables PLL band selection mode (0 = auto, and 1 = manual). | 1 |
| | | 5:0 | Manual VCO band | VCO band used in manual mode. | 0 |
| PLL Control 1 | 0C | 7:5 | PLL loop bandwidth | Selects PLL loop filter bandwidth.<br>000 = narrowest bandwidth.<br>…<br>111 = widest bandwidth. | 110 |
| | | 4:0 | 01001 | Set these bits to 01001 for optimal PLL operation. | 10001 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| PLL Control 2 | 0D | 7:6 | N2 | REFCLK-to-PLL controller clock rate ($f_{PC\_CLK}$).<br>00 = 2.<br>01 = 4.<br>10 = 8.<br>11 = 16.<br>$f_{PC\_CLK}$ must always be less than 50 MHz. | 11 |
| | | 4 | PLL cross control enable | Enables PLL cross-point control. | |
| | | 3:2 | N0 | VCO-to-DACCLK divider.<br>00 = 1.<br>01 = 2.<br>10 = 4.<br>11 = 4. | 001 |
| | | 1:0 | N1 | DACCLK-to-REFCLK divider.<br>00 = 2.<br>01 = 4.<br>10 = 8.<br>11 = 16. | 01 |
| PLL Status 0 | 0E | 3:0 | PLL control voltage | PLL VCO control voltage readback value. | Read-only |
| PLL Status 1 | 0F | 5:0 | VCO band readback | VCO band value. | Read-only |
| Sync Control 0 | 10 | 7 | Sync enable | Enables synchronization logic. | 0 |
| | | 6 | FIFO rate/data rate toggle | Operates synchronization at the FIFO reset rate (0)/data rate (1). | 0 |
| | | 3 | Rising edge sync | Rising edge of CLK samples sync input (1), falling edge of CLK samples sync input (0). | 1 |
| | | 2:0 | Sync averaging | Average sync input of number of samples.<br>000 = 1.<br>001 = 2.<br>010 = 4.<br>011 = 8.<br>100 = 16.<br>101 = 32.<br>110 = 64.<br>111 = 128. | 000 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| Sync Control 1 | 11 | 5:0 | Sync phase request | Offset of internal divided by 64 clock phase after sync.<br>000000 = 0 DAC clocks.<br>…<br>111111 = 63 DAC clocks. | 000000 |
| Sync Status 0 | 12 | 7 | Sync Lost | Synchronization lost. | Read-only |
| | | 6 | Sync locked | Synchronization found. | Read-only |
| Data Receiver Control | 14 | 6 | One DCI | 0 = two DCIs used, DCIA_x and DCIB_x.<br>1 = one DCI used, DCIA_x. | 0 |
| Data Receiver Status | 15 | 7 | LVDS receiver frame high | Frame input LVDS level > 1.7 V. | Read-only |
| | | 6 | LVDS receiver frame low | Frame input LVDS level < 0.7 V. | Read-only |
| | | 5 | LVDS receiver DCI high | DCI input LVDS level > 1.7 V. | Read-only |
| | | 4 | LVDS receiver DCI low | DCI input LVDS level < 0.7 V. | Read-only |
| | | 3 | LVDS receiver Port B high | Port B input LVDS level > 1.7 V. | Read-only |
| | | 2 | LVDS receiver Port B low | Port B input LVDS level < 0.7 V. | Read-only |
| | | 1 | LVDS receiver Port A high | Port A input LVDS level > 1.7 V. | Read-only |
| | | 0 | LVDS receiver Port A low | Port A input LVDS level < 0.7 V. | Read-only |
| FIFO Status/ Control Port A | 17 | 7 | FIFO Warning 1 | FIFO read and write pointers within ±1. | Read-only |
| | | 6 | FIFO Warning 2 | FIFO read and write pointers within ±2 | Read-only |
| | | 5 | FIFO reset aligned | FIFO read and write pointers aligned after chip reset. | Read-only |
| | | 4 | FIFO SPI align acknowledge | FIFO read and write pointers aligned after SPI driven FIFO reset. | Read-only |
| | | 3 | FIFO SPI align requesting | Request FIFO read and write pointers alignment via SPI. | 0 |
| | | 2:0 | FIFO phase offset | FIFO read and write pointer phase offset from optimal phase following FIFO reset.<br>000 = 0 offset from optimal phase.<br>…<br>111 = 7 offset from optimal phase.<br>The optimal value is 0. | 000 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| FIFO Status Port A | 18 | 7:0 | FIFO Level | Thermometer encoded measure of the FIFO level. | Read-only |
| FIFO Status/ Control Port B | 19 | 7 | FIFO Warning 1 | FIFO read and write pointers within ±1. | Read-only |
| | | 6 | FIFO Warning 2 | FIFO read and write pointers within ±2. | Read-only |
| | | 5 | FIFO reset aligned | FIFO read and write pointers aligned after chip reset. | Read-only |
| | | 4 | FIFO SPI align acknowledge | FIFO read and write pointers aligned after SPI driven FIFO reset. | Read-only |
| | | 3 | FIFO SPI align requesting | Request FIFO read and write pointers alignment via SPI. | 0 |
| | | 2:0 | FIFO phase offset | FIFO read and write pointer phase offset from optimal phase following FIFO reset.<br>000 = 0 offset from optimal phase.<br>…<br>111 = 7 offset from optimal phase.<br>The optimal value is 0. | 000 |
| FIFO Status Port B | 1A | 7:0 | FIFO level | Thermometer encoded measure of the FIFO level. | Read-only |
| HB1 Control | 1C | 7 | Enable pre mod | Enable fS/2 modulation stage that precedes Stage 1 interpolation filter. | 0 |
| | | 6 | Bypass sinc-1 | Sinc-1 filter bypass. | 1 |
| | | 2:1 | HB1[1:0] | Modulation mode for first stage interpolation filter ($f_{HB1} = 2 \times f_{IN1}$).<br>00 = input signal modulated by dc. Filter pass band is from −0.2 to +0.2 of $f_{HB1}$.<br>01 = input signal modulated by dc. Filter pass band is from 0.05 to 0.45 of $f_{HB1}$.<br>10 = input signal modulated by $f_{HB1}/2$. Filter pass band is from 0.3 to 0.7 of $f_{HB1}$.<br>11 = input signal modulated by $f_{HB1}/2$. Filter pass band is from 0.55 to 0.95 of $f_{HB1}$. | 00 |
| | | 0 | Bypass HB1 | First stage interpolation filter bypass. | 0 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| HB2 Control | 1D | | HB2[2:0] | Modulation mode for second stage interpolation filter ($f_{HB2} = 2 \times f_{IN2}$).<br>000 = input signal modulated by dc. Filter pass band is from −0.1 to +0.1 of $f_{HB2}$.<br>001 = input signal modulated by dc. Filter pass band is from 0.025 to 0.225 of $f_{HB2}$.<br>010 = input signal modulated by $f_{HB2}/4$. Filter pass band is from 0.15 to 0.35 of $f_{HB2}$.<br>011 = input signal modulated by $f_{HB2}/4$. Filter pass band is from 0.275 to 0.475 of $f_{HB2}$.<br>100 = input signal modulated by $f_{HB2}/2$. Filter pass band is from 0.4 to 0.6 of $f_{HB2}$.<br>101 = input signal modulated by $f_{HB2}/2$. Filter pass band is from 0.525 to 0.725 of $f_{HB2}$.<br>110 = input signal modulated by 3 $f_{HB2}/4$. Filter pass band is from 0.65 to 0.85 of fHB2.<br>111 = input signal modulated by 3 $f_{HB2}/4$. Filter pass band is from 0.775 to 0.975 of $f_{HB2}$. | 000 |
| | | 0 | Bypass HB2 | Second stage interpolation filter bypass. | 0 |
| HB3 Control | 1E | 7 | Bypass digital gain and phase adjustment | 1 = bypass digital gain and phase compensation. | 1 |
| | | 3:1 | HB3[2:0] | Modulation mode for third stage interpolation filter ($f_{HB3} = 2 \times f_{IN3}$).<br>000 = input signal modulated by dc. Filter pass band is from −0.1 to +0.1 of $f_{HB3}$.<br>001 = input signal modulated by dc. Filter pass band is from 0.025 to 0.225 of $f_{HB3}$.<br>010 = input signal modulated by $f_{HB3}/4$. Filter pass band is from 0.15 to 0.35 of $f_{HB3}$.<br>011 = input signal modulated by $f_{HB3}/4$. Filter pass band is from 0.275 to 0.475 of $f_{HB3}$.<br>100 = input signal modulated by $f_{HB3}/2$. Filter pass band is from 0.4 to 0.6 of $f_{HB3}$.<br>101: Input signal modulated by $f_{HB3}/2$. Filter pass band is from 0.525 to 0.725 of $f_{HB3}$.<br>110 = input signal modulated by 3 $f_{HB3}/4$. Filter pass band is from 0.65 to 0.85 of $f_{HB3}$.<br>111 = input signal modulated by 3 $f_{HB3}/4$. Filter pass band is from 0.775 to 0.975 of $f_{HB3}$. | 000 |
| | | 0 | Bypass HB3 | Third stage interpolation filter bypass. | 1 |
| Chip ID | 1F | 7:0 | Chip ID | Chip ID readback. | 20 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| Coeff I Byte 0 | 20 | 7 | 0 | Set this bit to 0. | 0 |
| | | 6:3 | Coeff_1i[3:0] | I-Path DAC Sinc$^{-1}$ Filter Coefficient 2 in twos complement format. | 0 |
| | | 2:0 | Coeff_0i | I-Path DAC Sinc$^{-1}$ Filter Coefficient 1 in twos complement format.<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | 0 |
| Coeff I Byte 1 | 21 | 7:5 | Coeff_3i[2:0] | I-Path DAC Sinc$^{-1}$ Filter Coefficient 4 (LSB) in twos complement format. | 6 |
| | | 4:0 | Coeff_2i | I-Path DAC Sinc$^{-1}$ Filter Coefficient 3 in twos complement format.<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | 0 |
| Coeff I Byte 2 | 22 | 7:5 | Coeff_4i[2:0] | I-Path DAC Sinc$^{-1}$ Filter Coefficient 5 (LSB) in twos complement format. | 7 |
| | | 4 | 0 | Set this bit to 0. | 0 |
| | | 3:0 | Coeff_3i[6:3] | Set I-Path DAC Sinc$^{-1}$ Filter Coefficient 4 (MSB) in twos complement format.<br>DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | F |
| Coeff I Byte 3 | 23 | 7 | 0 | Set this bit to 0. | 0 |
| | | 6:0 | Coeff_4i[9:3] | I-Path DAC Sinc$^{-1}$ Filter Coefficient 5 (MSB) in twos complement format.<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | 7F |
| Coeff Q Byte 0 | 24 | 7 | 0 | Set this bit to 0. | 0 |
| | | 6:3 | Coeff_1q[3:0] | Q-Path DAC Sinc$^{-1}$ Filter Coefficient 2 in twos complement format. | D |
| | | 2:0 | Coeff_0q | Q-Path DAC Sinc$^{-1}$ Filter Coefficient 1 in twos complement format.<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 1 |
| Coeff Q Byte 1 | 25 | 7:5 | Coeff_3q[2:0] | Q-Path DAC Sinc$^{-1}$ Filter Coefficient 4 (LSB) in twos complement format. | 7 |
| | | 4:0 | Coeff_2q | Q-Path DAC Sinc$^{-1}$ Filter Coefficient 3 in twos complement format.<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 6 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| Coeff Q Byte 2 | 26 | 7:5 | Coeff_4q[2:0] | Q-Path DAC Sinc$^{-1}$ Filter Coefficient 5 (LSB) in twos complement format. | 0 |
| | | 4 | 0 | Set this bit to 0. | 0 |
| | | 3:0 | Coeff_3q[6:3] | Q-Path DAC Sinc$^{-1}$ Filter Coefficient 4 (MSB) in twos complement format.<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | D |
| Coeff Q Byte 3 | 27 | 7 | 0 | Set this bit to 0. | 0 |
| | | 6:0 | Coeff_4q[9:3] | Q-Path DAC Sinc$^{-1}$ Filter Coefficient 5 (MSB) in twos complement format.<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 0 |
| I Phase Adj LSB | 28 | 7:0 | Phase Word I[7:0] | See Register 0x29. | 0 |
| I Phase Adj MSB | 29 | 1:0 | Phase Word I[9:8] | Phase Word I[9:0] is used to insert a phase offset between the I and Q data paths. The adjustment range is ±1.75°.<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | 0 |
| Q Phase Adj LSB | 2A | 7:0 | Phase Word Q[7:0] | See Register 0x2B. | 0 |
| Q Phase Adj MSB | 2B | 1:0 | Phase Word Q[9:8] | Phase Word Q[9:0] is used to insert a phase offset between the I and Q data paths. The adjustment range is ±1.75°.<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 0 |
| I DC Offset LSB | 2C | 7:0 | DC Offset I[7:0] | See Register 0x2D. | 0 |
| I DC Offset MSB | 2D | 7:0 | DC Offset I[15:8] | DC Offset I[15:0] is a value added directly to the samples written to the IDAC. The LSB bit weight is $2^0$. The adjustment range is ±10 mA.<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | 0 |
| Q DC Offset LSB | 2E | 7:0 | DC Offset Q[7:0] | See Register 0x2F. | 0 |
| Q DC Offset MSB | 2F | 7:0 | DC Offset Q[15:8] | DC Offset Q[15:0] is a value added directly to the samples written to the QDAC. The LSB bit weight is $2^0$. The adjustment range is ±10 mA.<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 0<br>0 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| IDAC FSC Adj | 30 | 7:0 | IDAC FSC Adj | IDAC full-scale current adjustment (LSB part). IDAC FS Adj[9:0] sets the full-scale current of the IDAC. The full-scale current can be adjusted from 8.64 mA to 31.6 mA in step sizes of approximately 22.5 µA.<br>0x000 = 8.64 mA.<br>...<br>0x200 = 20.14 mA.<br>…<br>0x3FF = 31.66 mA.<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | F9 |
| IDAC Control | 31 | 7 | IDAC sleep | I DAC sleep mode (fast wake-up mode). | 0 |
| | | 1:0 | IDAC FSC Adj[9:8] | IDAC full-scale current adjustment (MSB part)<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | 01 |
| Aux IDAC Data | 32 | 7:0 | AUX IDAC Data | Auxiliary IDAC data (LSB part). AUX IDAC Data[9:0] sets the magnitude of the aux DAC current. The range is 0 mA to 2 mA, and the step size is 2 µA.<br>0x000 = 0.000 mA.<br>0x001 = 0x002 mA.<br>…<br>0x3FF = 2.046 mA.<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | 00 |
| Aux IDAC Control | 33 | 7 | AUX IDAC sign | Auxiliary IDAC output sign.<br>0 = positive, current is directed to the AUXx_P pin.<br>1 = negative, current is directed to the AUXx_N pin. | 0 |
| | | 6 | AUX IDAC current direction | Auxiliary IDAC current direction.<br>0 = source.<br>1 = sink. | 0 |
| | | 5 | AUX IDAC power-down | Auxiliary IDAC power-down. | 0 |
| | | 1:0 | AUX IDAC Data[9:8] | Auxiliary IDAC data (MSB part).<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select =1 to configure DAC 3 path. | 00 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| QDAC FSC Adj | 34 | 7:0 | QDAC FSC Adj | Q DAC full-scale current adjustment (LSB part). QDAC FS Adj[9:0] sets the full-scale current of the QDAC. The full-scale current can be adjusted from 8.64 mA to 31.6 mA in step sizes of approximately 22.5 µA.<br>0x000 = 8.64 mA<br>...<br>0x200 = 20.14mA<br>…<br>0x3FF = 31.66 mA<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | F9 |
| QDAC Control | 35 | 7 | QDAC sleep | Q DAC sleep mode (fast wake-up mode). | 0 |
| | | 1:0 | QDAC FSC Adj[9:8] | QDAC full-scale current adjustment (MSB part).<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 01 |
| Aux QDAC Data | 36 | 7:0 | AUX QDAC Data | Auxiliary QDAC data (LSB part). AUX QDAC Data[9:0] sets the magnitude of the AUX DAC current. The range is 0 mA to 2 mA, and the step size is 2 µA.<br>0x000 = 0.000 mA.<br>0x001 = 0x002 mA.<br>…<br>0x3FF = 2.046 mA.<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 00 |
| Aux QDAC Control | 37 | 7 | AUX QDAC sign | Auxiliary QDAC output sign.<br>0 = positive, current is directed to the AUXx_P pin.<br>1 = negative, current is directed to the AUXx_N pin. | 0 |
| | | 6 | AUX QDAC current direction | Auxiliary QDAC current direction.<br>0 = source.<br>1 = sink. | 0 |
| | | 5 | AUX QDAC power-down | Auxiliary QDAC power-down. | 0 |
| | | 1:0 | AUX QDAC Data[9:8] | Auxiliary QDAC data (MSB part).<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 00 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| SED_S0_L | 38 | 7:0 | SED Compare Pattern Sample0[7:0] | Compare Pattern Sample0[15:0] is the word that is compared with Data Sample 0 captured at the input interface by the rising edge of DCI.<br>Set DAC SPI select = 0 to configure Port A.<br>Set DAC SPI select = 1 to configure Port B. | |
| SED_S0_H | 39 | 7:0 | SED Compare Pattern Sample0[15:8] | Compare Pattern Sample0[15:0] is the word that is compared with Data Sample 0 captured at the input interface by the rising edge of DCI.<br>Set DAC SPI select = 0 to configure Port A.<br>Set DAC SPI select = 1 to configure Port B. | |
| SED_S1_L | 3A | 7:0 | SED Compare Pattern Sample1[7:0] | Compare Pattern Sample1[15:0] is the word that is compared with Data Sample 1 captured at the input interface by the falling edge of DCI.<br>Set DAC SPI select = 0 to configure Port A.<br>Set DAC SPI select = 1 to configure Port B. | |
| SED_S1_H | 3B | 7:0 | SED Compare Pattern Sample1[15:8] | Compare Pattern Sample1[15:0] is the word that is compared with Data Sample 1 captured at the input interface by the falling edge of DCI.<br>Set DAC SPI select = 0 to configure Port A.<br>Set DAC SPI select = 1 to configure Port B. | |
| SED_S2_L | 3C | 7:0 | SED Compare Pattern Sample2[7:0] | Compare Pattern Sample2[15:0] is the word that is compared with Data Sample 2 captured at the input interface by the rising edge of DCI.<br>Set DAC SPI select = 0 to configure Port A.<br>Set DAC SPI select = 1 to configure Port B. | |
| SED_S2_H | 3D | 7:0 | SED Compare Pattern Sample2[15:8] | Compare Pattern Sample2[15:0] is the word that is compared with Data Sample 2 captured at the input interface by the rising edge of DCI.<br>Set DAC SPI select = 0 to configure Port A.<br>Set DAC SPI select = 1 to configure Port B. | |
| SED_S3_L | 3E | 7:0 | SED Compare Pattern Sample3 [7:0] | Compare Pattern Sample3[15:0] is the word that is compared with Data Sample 3 captured at the input interface by the falling edge of DCI.<br>Set DAC SPI select = 0 to configure Port A.<br>Set DAC SPI select = 1 to configure Port B. | |
| SED_S3_H | 3F | 7:0 | SED Compare Pattern Sample3[15:8] | Compare Pattern Sample3[15:0] is the word that is compared with Data Sample 3 captured at the input interface by the falling edge of DCI.<br>Set DAC SPI select = 0 to configure Port A.<br>Set DAC SPI select = 1 to configure Port B. | |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| SED Control/Status | 40 | 7 | SED compare enable | Enables the SED circuitry. | 0 |
| | | 6 | Port B error detected | Status of last compare on Port B. | 0 |
| | | 5 | Port A error detected | Status of last compare on Port A. | 0 |
| | | 3 | Auto-clear enable | Enables the auto reset after eight valid sample sets. | 0 |
| | | 2 | Port B compare failed | Fail status determined for last sample set on Port B. | 0 |
| | | 1 | Port A compare failed | Fail status determined for last sample set on Port A. | 0 |
| | | 0 | Compare passed | Pass status determined for last sample set. | 0 |
| SED_R_L | 41 | 7:0 | SED Status Rising Edge Samples[7:0] | SED Status Rising Edge Samples[15:0] indicate which bits were received in error.<br>Set DAC SPI select = 0 to read back errors on Port A.<br>Set DAC SPI select = 1 to read back errors on Port B. | Read-only |
| SED_R_H | 42 | 7:0 | SED Status Rising Edge Samples[15:8] | SED Status Rising Edge Samples[15:0] indicate which bits were received in error.<br>Set DAC SPI select = 0 to read back errors on Port A.<br>Set DAC SPI select = 1 to read back errors on Port B. | Read-only |
| SED_F_L | 43 | 7:0 | SED Status Falling Edge Samples[7:0] | SED Status Falling Edge Samples[15:0] indicate which bits were received in error.<br>Set DAC SPI select = 0 to read back errors on Port A.<br>Set DAC SPI select = 1 to read back errors on Port B. | Read-only |
| SED_F_H | 44 | 7:0 | SED Status Falling Edge Samples[15:8] | SED Status Falling Edge Samples[15:0] indicate which bits were received in error.<br>Set DAC SPI select = 0 to read back errors on Port A.<br>Set DAC SPI select = 1 to read back errors on Port B. | Read-only |
| I Gain Control | 50 | 7:0 | IGain | IGain[7:0] is a value that directly scales the samples written to the IDAC. The bit weighting is MSB = $2^1$ and LSB = $2^{-6}$, which yields a multiplier range of 0 to 3.984375.<br>Set DAC SPI select = 0 to configure DAC 1 path.<br>Set DAC SPI select = 1 to configure DAC 3 path. | 40 |
| Q Gain Control | 51 | 7:0 | QGain | QGain[7:0] is a value that directly scales the samples written to the QDAC. The bit weighting is MSB = $2^1$ and LSB = $2^{-6}$, which yields a multiplier range of 0 to 3.984375.<br>Set DAC SPI select = 0 to configure DAC 2 path.<br>Set DAC SPI select = 1 to configure DAC 4 path. | 40 |

| Register Name | Addr (Hex) | Bit | Name | Function | Default |
|---|---|---|---|---|---|
| FTW (LSB) | 54 | 7:0 | FTW[7:0] | See Register 0x57. | 0 |
| FTW | 55 | 7:0 | FTW[15:8] | See Register 0x57. | 0 |
| FTW | 56 | 7:0 | FTW [23:16] | See Register 0x57. | 0 |
| FTW (MSB) | 57 | 7:0 | FTW [31:24] | FTW[31:0] is the 32-bit frequency tuning word that determines the frequency of the complex carrier generated by the on-chip NCO. The frequency is not updated when the FTW registers are written. The values are only updated when Register 0x5A[2] transitions from 0 to 1. | 0 |
| Phase Offset MSB | 58 | 7:0 | NCO Phase Offset[15:8] | See Register 0x59. | 0 |
| Phase Offset LSB | 59 | 7:0 | NCO Phase Offset[7:0] | NCO Phase Offset[15:0] sets the phase of the complex carrier signal when the NCO is reset. The phase offset spans between 0º and 360º. Each bit represents an offset of 0.0055°. Value is in twos complement format. | 0 |
| DDS/Mod Control | 5A | 7 | Bypass DDS/MOD | 1 = bypass NCO. | 1 |
| | | 5 | Frame NCO reset ack | 1 = indicates that the NCO has been reset due to an extended FRAMEx_x pulse signal. | 0 |
| | | 4 | Frame NCO reset request | 0→1 = The NCO is reset on the first extended FRAMEx_x pulse after this bit transitions from 0 to 1. | 0 |
| | | 3 | FTW update ack | 1 = indicates that the FTW has been updated with the SPI value. | 0 |
| | | 2 | FTW update request | 0→1 = FTW is updated with the SPI value on a 0-to-1 transition of this bit. | 0 |
| | | 0 | Sideband select | 0 = The modulator output high-side image.<br>1 = The modulator output low-side image. The image is spectrally inverted compared to the input data. | 0 |
| Die Temp Control 0 | 5C | 1 | Latch temp data | 0 → 1 = latches temp sensor data. This should be completed before the Die Temp[15:0] is readback. | 0 |
| | | 0 | Temp sensor power-down | 1 = powers down the aux ADC that converts die temperature. | 1 |
| Die Temp Control 1 | 5D | 7:0 | 00001010 | Set these bits to 00001010 for optimal temperature sensor operation. | 100000 |
| Die Temp (LSBs) | 5E | 7:0 | Die Temp[7:0] | Die Temp[15:0] indicates the approximate die temperature. | Read-only |
| Die Temp (MSBs) | 5F | 7:0 | Die Temp[15:8] | Die Temp[15:0] indicates the approximate die temperature. | Read-only |
| DCI Delay | 72 | 1:0 | DCI delay | Programmable delay added DCI.<br>00 = no added delay.<br>01 = 200 ps delay.<br>10 = 400 ps delay.<br>11 = 600 ps delay. | 00 |

# INPUT DATA PORTS

The AD9148 can operate in three data input modes: dual-port mode, single-port mode, and byte mode. In dual-port mode, DAC 1 and DAC 2 receive data from Port A, and DAC 3 and DAC 4 receive data from Port B. In single-port mode, all four DACs receive data from Port A. In byte mode, all four DACs receive data from Port A, but the port is split into two 8-bit wide buses. In all modes, the data input timing is relative to a DCI signal provided with the data.

## DUAL-PORT MODE

In dual-port mode, the DCI signal indicates to which DAC the data is intended. On the rising edge of DCI, data is latched into DAC 1 and DAC 3. On the falling edge of DCI, data is latched into DAC 2 and DAC 4. This pattern repeats continuously.

There is a SPI programmable option (Register 0x14[6]) to provide one DCI for both input ports or two DCIs, where each DCI is associated with one input port. Two DCIs are useful when the data for each port is coming from a different data source. These cases are illustrated in Figure 45 and Figure 46.

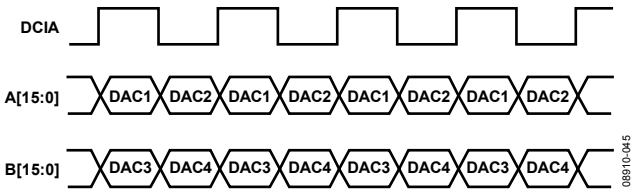



*Figure 45. Timing Diagram for Dual-Port Mode, One DCI*

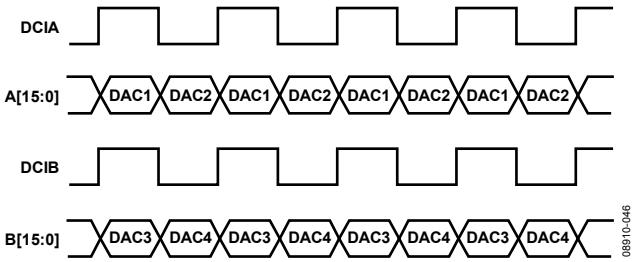



*Figure 46. Timing Diagram for Dual-Port Mode, Two DCI*

Each data sample, by default, is expected to be formatted as an MSB sent to Bit 15 and an LSB sent to Bit 0 for each port. The AD9148 contains an option to swap the bus (Register 0x03[4]). When this bus swap bit is set, the MSB should be sent to Bit 0, and the LSB should be sent to Bit 15 for each port.

## SINGLE-PORT MODE

In single-port mode, a FRAME signal must be provided along with the DCI signal and the data. The FRAME signal indicates to which DAC the data is intended. When FRAME goes high, the first data-word goes to DAC 1, and the second data-word goes to DAC 2. When FRAME goes low, the first data-word goes to DAC 3, and the second data-word goes to DAC 4. This pattern repeats continuously as illustrated in Figure 47.

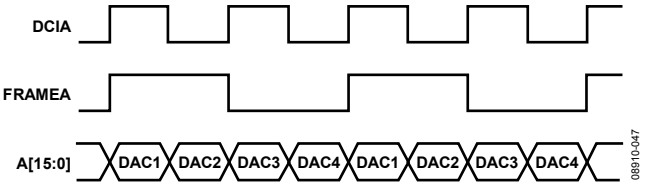



*Figure 47. Timing Diagram for Single-Port Mode*

Each data sample, by default, is expected to be formatted as an MSB sent to Bit 15 and an LSB sent to Bit 0. When the bus swap bit is set (Register 0x03[4]), the MSB should be sent to Bit 0, and the LSB should be sent to Bit 15 for each port.

The FRAME signal is sampled with the same internal signal as the data and has the same set-up and hold timing relative to DCI. If desired, only the first FRAME pulse needs to be generated. This initializes the internal clock phases inside the device, and data latches just as if the periodic FRAME signal were sent.

## BYTE MODE

In byte mode, a FRAME signal must be provided along with the DCI signal and the data. The most significant byte of the data should correspond with DCI being high, and the least significant byte of the data should correspond with DCI being low. The FRAME signal indicates to which DAC the data is intended. When FRAME is high, data on the top half of the port (A[15:8]) is sent to DAC 1 and data on the bottom half of the port (A[7:0]) is sent to DAC 3. When the FRAME is low, data on the top half of the port is sent to DAC 2 and data on the bottom half of the port is sent to DAC 4. This pattern repeats continuously as shown in Figure 48.

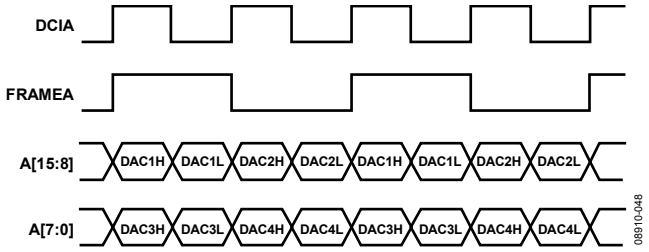



*Figure 48. Timing Diagram for Byte Mode*

The AD9148 also includes a byte swap feature. By default, the bytes should be formatted as an MSB sent to Bit 15 on Bus 1 and Bit 7 on Bus 2. When byte swap is enabled (Register 0x03[2]), an MSB should be sent to Bit 8 on Bus 1 and Bit 0 on Bus 2. This is described in Table 14.

**Table 14. Byte Swap Formatting**

| Byte Swap | Byte | A[15:8] | A[7:0] |
|---|---|---|---|
| 0 | MSB | Data Set 1[15:8] | Data Set 2[15:8] |
| 0 | LSB | Data Set 1[7:0] | Data Set 2[7:0] |
| 1 | MSB | Data Set 1[8:15] | Data Set 2[8:15] |
| 1 | LSB | Data Set 1[0:7] | Data Set 2[0:7] |

## DATA INTERFACE OPTIONS

To enable optimization of the data interface, some additional options have been provided in the following registers:

- Data format (Register 0x03)
- Data receiver control (Register 0x14)
- Data receiver status (Register 0x15)

Depending on the data rate and DCI vs. data skew, the internal DCI can be inverted to meet the valid data timing window.

## RECOMMENDED FRAME INPUT BIAS CIRCUITRY

Because the frame signal can be used as a reference clock in the byte mode or as a trigger to reset the FIFO, it is recommended that the frame input be tied to LVDS logic low when it is not used (that is, when it is not driven by an ASIC or FPGA). The external bias circuit shown in Figure 49 is recommended for this purpose. This bias circuit applies to both FRAMEA and FRAMEB ports.

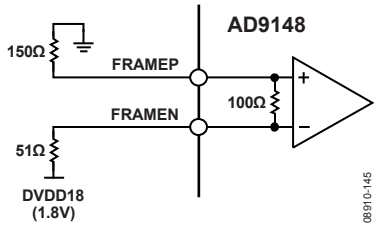



*Figure 49. External Bias Circuit*

# FIFO OPERATION

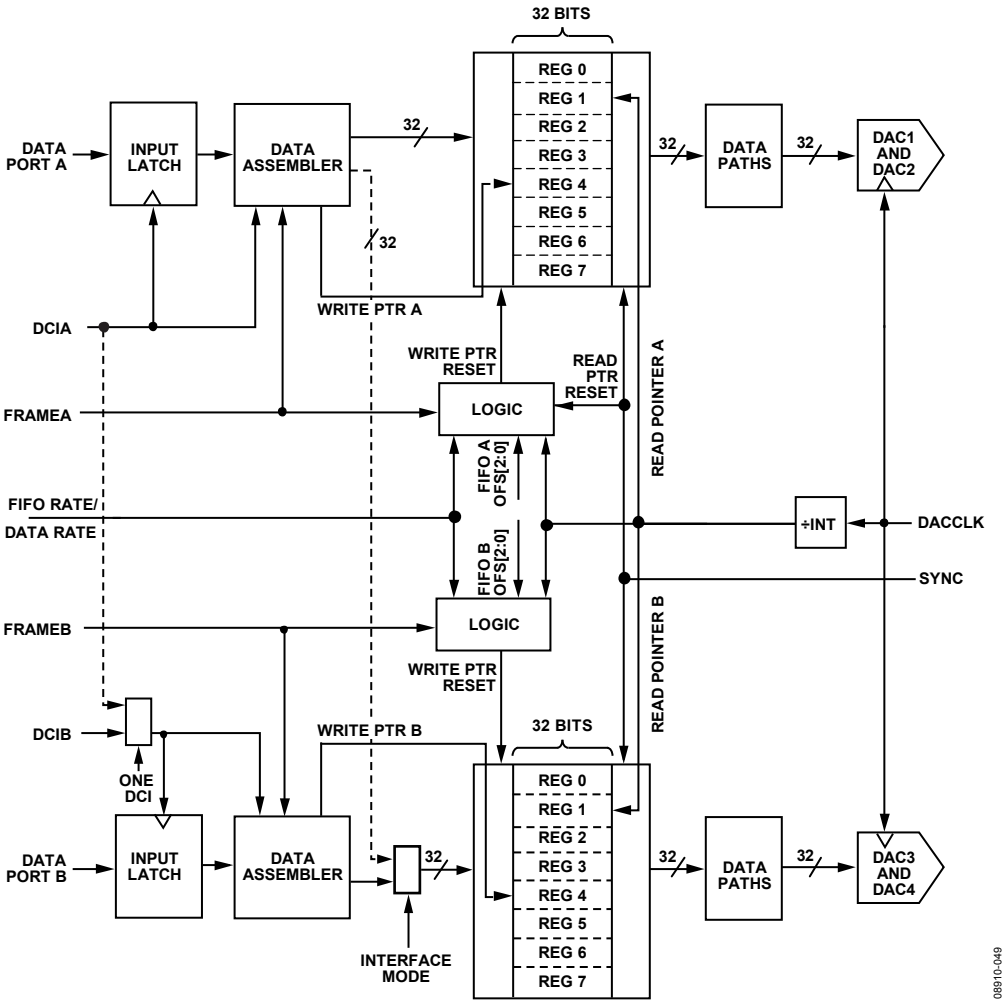



*Figure 50. Block Diagram of FIFO*

The AD9148 contains two 32-bit wide, 8-word deep FIFOs (one per dual DAC) designed to relax the timing relationship between the data arriving at the DAC input ports and the internal DAC data rate clock. The FIFOs can also be used to provide an adjustable pipeline delay between the DCIx clocks and the DACCLK allowing realignment of data input in a multichip system. This significantly increases the timing budget of the interface.

Figure 50 shows the block diagram of the datapath through the FIFO. The data is latched into the device, is formatted, and is then written into the FIFO register determined by the FIFO write pointer. The value of the write pointer is incremented every time a new word is loaded into the FIFO. Meanwhile, data is read from the FIFO register determined by the read pointer and fed into the digital datapath. The value of the read pointer is updated every time data is read into the datapath from the FIFO. This happens at the data rate, that is the DACCLK rate divided by the interpolation ratio. The difference between the write and read pointers represents the FIFO pipeline delay and is important to take into account when understanding the overall pipeline delay of the AD9148.

In single-port and byte interface modes, the incoming digital data is sampled at twice the data rate (DCIA). The data is then assembled based on the interface mode. At the output of the data assembler block, the data samples for DAC 1 and DAC 2 are written to FIFO A and the data samples for DAC 3 and DAC 4 are written to FIFO B at the data rate.

Valid data is transmitted through the FIFO as long as the FIFO does not overflow or become empty. An overflow or empty condition of the FIFO is the same as the write pointer and the read pointer being equal. When both pointers are equal, an attempt is made to read and write a single FIFO register simultaneously. This simultaneous register access leads to unreliable data transfer through the FIFO and must be avoided.

Nominally, data is written to the FIFO at the same rate as data is read from the FIFO. This keeps the data level in the FIFO constant. If data is written to the FIFO faster than data is read, the data level in the FIFO increases. If the data is written to the device slower than data is read, the data level in the FIFO decreases. For a maximum timing margin, the FIFO level should be maintained near half full, which is the same as maintaining a difference of 4 between the write pointer and read pointer values.

## SYNCHRONIZING AND RESETTING THE FIFO

To avoid any concurrent reads and writes to the same FIFO address and to assure a fixed pipeline delay, it is important to reset the state of the FIFO pointers to known states. The pipeline delay in the AD9148 comes from two sources, FIFO delay and the delay though the signal processing in the DAC.

To assure a fixed and predictable pipeline delay in the signal processing, the FIFO read operation is synchronized with the DACCLK and, more importantly, in case of interpolation, its divided down version so that the same edge of the slowest clock in the signal processing reads the same data in the FIFO. The synchronization is performed by resetting the FIFO read pointer to a known state relative to the slowest clock used in the signal processing. This synchronization is enabled by setting Bit 7 in Register 0x10 to 1, and it uses the REFCLK/SYNC signal for its reference.

To manage the FIFO pipeline delay, the FIFO write pointer must be synchronized with the read pointer to avoid concurrent access to the FIFO and to potentially compensate for any data input phase mismatch. This synchronization can be performed either at the data rate (see the Data Rate Synchronization section) or at the FIFO rate (see the FIFO Rate Synchronization section).

### *FIFO Synchronization Modes*

To benefit from the advantages of the FIFO functionality in the different modes of operations, PLL on/off, standalone, or multi-chip synchronization, the FIFO can operate in the following ways:

- Synchronization at the data rate
- Synchronization at the FIFO rate (data rate/FIFO depth)
- No synchronization

As discussed in the Input Data Ports section, in single-port mode and byte mode, the FRAME input is used as a data select signal that indicates to which DAC the input data is intended to be written. When synchronization is needed, the FRAME signal is given another function, initializing the FIFO write pointer address. When the FRAME signal is asserted high for at least the time interval needed to load complete data to the four DACs (which correspond to one DCI period in dual-port mode and two DCI periods in single-port mode or byte mode), the FIFO write pointer is reset to a value dependent on the synchronization mode selected and the FIFO phase offset bits of the corresponding FIFO Status/Control Port x register, Register 0x17 or Register 0x19.

### Data Rate Synchronization

In this mode, the REFCLK/SYNC signal is used to reset the FIFO read pointer to 0. The edge of the CLK used to sample the SYNC signal is selected by Bit 3 of Register 0x10. If the PLL is used, REFCLK is used as a SYNC signal, and the FIFO read pointer is reset at the REFCLK rate divided by 64. The data rate synchronization is selected by setting Bit 6 of Register 0x10 to 0.

As previously mentioned, the FRAME signal is used to reset the FIFO write pointer. When the FRAME is asserted, the FIFO write pointer is reset to the address defined in Bits[2:0] of the corresponding FIFO Status/Control Port x register (Register 0x17 or Register 0x19) the next time the read pointer becomes 0 (see Figure 51).

The data rate synchronization, the write pointer of the FIFO, and the read pointer of the FIFO are synchronized at the SYNC rate and have a fixed phase offset.

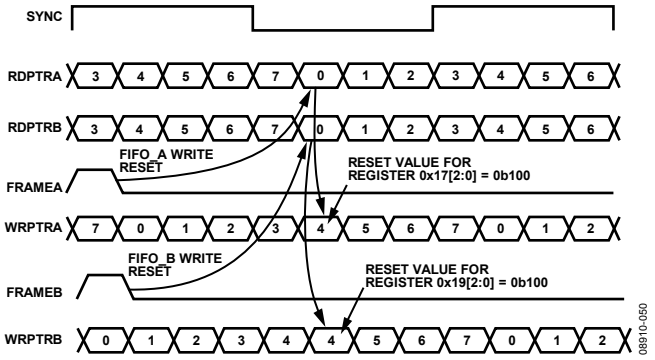



*Figure 51. Timing of the FRAME Input vs. Write Pointer Value in Data Rate Synchronization*

### FIFO Rate Synchronization

In this mode, the REFCLK/SYNC signal is used to reset the FIFO read pointer to 0. The edge of the CLK_x used to sample the SYNC signal is selected by Bit 3 of Register 0x10. As previously mentioned, the FRAME signal is used to reset the FIFO write pointer. In the FIFO rate synchronization mode, the FIFO write pointer is reset immediately after the FRAME signal is asserted high for at least the time interval needed to load complete data to the four DACs, and the FIFO write pointer is reset to the address defined in Bits[2:0] of the corresponding FIFO Status/Control Port x register, Register 0x17 or Register 0x19 (see Figure 52).

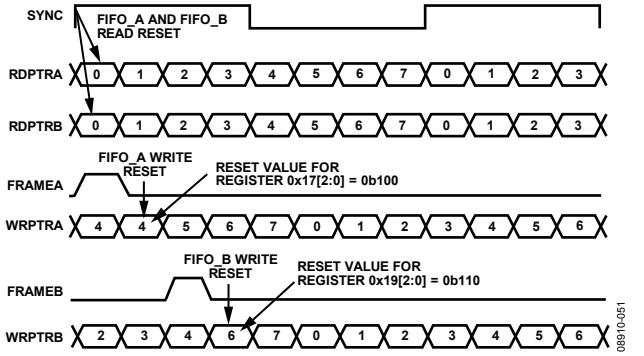



*Figure 52. Timing of the FRAME Input vs. Write Pointer Value in FIFO Rate Synchronization*

**No Synchronization**

In this mode, Bit 7 in Register 0x10 is set to 0, the pipeline delay in the signal processing is not controlled, and the read pointer of the FIFO is never reset. However, to assure that the FIFO can operate safely and there is no concurrent access to FIFO from the write and read pointer to the same address, it is important to ensure that the phase offset between the two pointers is greater than 2. In consequence, the only FIFO reset that can be used safely is the data rate synchronization, Bit 6 of Register 0x10 set to 0, where the FIFO is reset with a fixed offset of 4 between the write and read pointers. Because there is no SYNC signal, the reset of the FIFO write pointer can only be done by a FRAME signal or an SPI command.

### *FIFO Reset Commands*

Depending on the configuration of the system, the FIFO reset can be done manually or periodically for a multichip system. The AD9148 provides two ways to resetting the FIFO pointers: SPI interface or periodic reset using the FRAME signal.

The SPI also gives access to each FIFO phase offset in Bits [2:0] of the corresponding FIFO status/control registers, Register 0x17 and Register 0x19. The value in these three bits corresponds either to the offset between the write and read pointer in the data rate synchronization or to the absolute address of the FIFO write pointer in the FIFO rate synchronization.

**SPI Command for Manual Reset**

If a manual reset is acceptable, the FIFO pointer addresses can be reset using the SPI interface.

To initialize the FIFO data level through the SPI, Bit 3 of Register 0x17 (FIFO Port A) or Bit 3 of Register 0x19 (FIFO Port B) should be toggled from 0 to 1 and back. When the write to the register is complete, the corresponding FIFO data level is initialized.

The recommended procedure for a SPI FIFO data level initialization is

1. Request FIFO Port A or FIFO Port B level reset by setting Bit 3 in Register 0x17 or Bit 3 in Register 0x19 to Logic 1. The FIFO phase offset, Bits [2:0] in Register 0x17 or Bits [2:0] in Register 0x19, should also be written at the same time to set the desired value of offset between the FIFO write and read pointers.
2. Verify that the part acknowledges the request by ensuring that Bit 4 in Register 0x17 or Bit 4 in Register 0x19 is set to Logic 1.
3. Remove the request by resetting Bit 3, Register 0x17 or Bit 3, Register 0x19 to 0.
4. The FIFO SPI aligned flag in the Event Flag 0 register, Bit 2 in Register 0x06, is set when the reset of the write pointer has been realized. Bit 4 in Register 0x17 or Bit 4 in Register 0x19 is reset to 0 to indicate which FIFO has generated this flag.

Note that the SPI writes to Register 0x17 or Register 0x19 should be done while maintaining a constant value in the FIFO phase offset bits.

**FIFO Reset Using FRAME Signal**

The FIFO pointers can also be reset using the FRAME signals. If only one DCI is used, only the FRAMEA signal is used for the FIFO reset. This mode is enabled by setting Bit 6 in Register 0x10.

As discussed in the FIFO Synchronization Modes section, the FRAME input is used to initialize the FIFO data level value. When the FRAME signal is asserted high for at least the time interval needed to load the complete data to the four DACs, the write pointer is reset depending on the mode of synchronization chosen.

- Data rate synchronization (default), Bit 6 of Register 0x10, is set to 0. When read pointer reaches 0, write pointer reset to FIFO offset phase.
- FIFO rate synchronization, Bit 6 of Register 0x10, is set to 1. On the rising edge of the FRAME signal, write pointer reset to FIFO start level.

## MONITORING THE FIFO STATUS

The FIFO initialization and status can be read from Register 0x17. This register provides information about the FIFO initialization method and whether the initialization was successful. The MSB of Register 0x17 is a FIFO warning flag that can optionally trigger a device $\overline{\text{IRQ}}$. This flag is an indication that the FIFO is close to emptying (FIFO level is 1) or overflowing (FIFO level is 7). This is an indication that the data may soon be corrupted, and action should be taken.

The FIFO data level can be read from Register 0x18 at any time. The SPI reported FIFO data level is denoted as a 7-bit thermometer code of the write counter state relative to the absolute read counter being 0. The optimum FIFO data level of four is, therefore, reported as a value of 00001111 in the status register.

Note that, depending on the timing relationship between DCI and the main DACCLK, the FIFO level value can be off by a ±1 count. Therefore, it is important to keep the difference between the read and write points to at least 2.

# DEVICE SYNCHRONIZATION

## SYNCHRONIZING MULTIPLE DEVICES

System demands may require that the outputs of multiple DACs be synchronized with each other or with a system clock. Systems that support transmit diversity or beam-forming, where multiple antennas are used to transmit a correlated signal, require multiple DAC outputs to be phase aligned with each other. Systems with a time-division multiplexing transmit chain may require one or more DACs to be synchronized with a system-level reference clock.

Multiple devices are considered synchronized to each other when the state of the clock generation state machines is identical for all parts and time aligned data is being read from the FIFOs of all parts simultaneously. Devices are considered synchronized to a system clock when there is a fixed and known relationship between the clock generation state machine and the data being read from the FIFO and a particular clock edge of the system clock. The AD9148 has provisions for enabling multiple devices to be synchronized to each other or to a system clock.

The AD9148 supports synchronization in two different modes, data rate mode and FIFO rate mode. The two modes are distinguished by the lowest rate clock that the synchronization logic attempts to synchronize. In data rate mode, the input data rate represents the lowest synchronized clock. In FIFO rate mode, the FIFO rate, which is the data rate divided by the FIFO depth of 8, represents the lowest rate clock. The advantage of the FIFO rate synchronization is increased setup and hold times of DCI relative to the CLK input. When in data rate synchronization mode, the elasticity of the FIFO is not used to absorb timing variations between the data source and DAC, resulting in tighter setup and hold time requirements.

The method chosen for providing the DAC sampling clock directly impacts the synchronization methods available. When the device clock multiplier is used, only data rate synchronization is available. When the DAC sampling clock is sourced directly, both data rate mode and FIFO rate mode synchronization are available.

## SYNCHRONIZATION WITH CLOCK MULTIPLICATION

When using the clock multiplier to generate the DACCLK, the REFCLK/SYNC input signal acts as both the reference clock for the PLL-based clock multiplier and as the synchronization signal. To synchronize devices, the REFCLK/SYNC signal must be distributed with low skew to all of the devices to be synchronized. Skew between the REFCLK/SYNC signals of different devices show up directly as a timing mismatch at the DAC outputs.

The frequency of the REFCLK/SYNC signal is typically equal to the input data rate. The FRAME signal and DCI signals can be created in the FPGA along with the data. A circuit diagram of a typical configuration is shown in Figure 53.

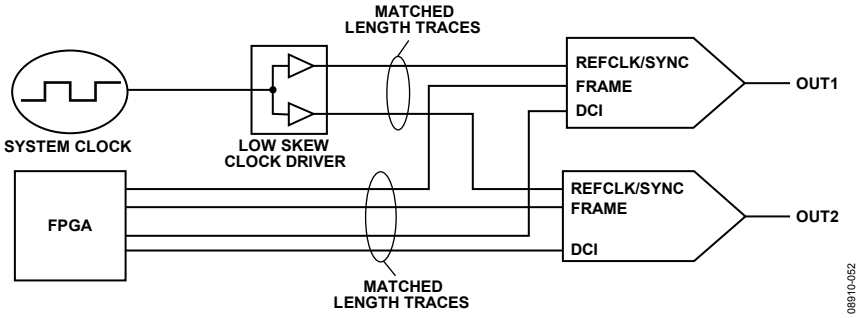



*Figure 53. Typical Circuit Diagram for Synchronizing Devices with Clock Multiplication Enabled*

The following procedure outlines the steps required to synchronize multiple devices. The procedure assumes that the REFCLK/SYNC signal is applied to all of the devices and the PLL of each device is phase locked to it. Each individual device must follow this procedure.

The procedure for synchronization when using the PLL follows:

1. Configure for data rate, periodic synchronization by writing 0xC0 to the sync control register (Register 0x10).
2. Read the sync status register (Register 0x12) and verify that the sync locked bit (Bit 6) is set high indicating that the device achieved back-end synchronization and that the sync lost bit (Bit 7) is low. These levels indicate that the clocks are running with a constant and known phase relative to the sync signal.
3. Reset the FIFO by strobing the FRAME signal high for at least the time interval needed to load complete data to the four DACs. Resetting the FIFO ensures that the correct data is being read from the FIFO. This completes the synchronization procedure, and at this stage, all devices should be synchronized.

To maintain synchronization, the skew between REFCLK/SYNC signals of the devices must be less than $t_{SKEW}$ nanoseconds. There is also a setup and hold time to be observed between the DCI and data of each device and the REFCLK/SYNC signal. When resetting the FIFO, the FRAME signal must be held high for at least the time interval needed to load complete data to the four DACs (one DCI period for dual-port mode and two DCI periods for single-port or byte mode). A timing diagram of the input signals is shown in Figure 54.

The example in Figure 54 shows a REFCLK/SYNC frequency equal to the data rate. Whereas this is the most common situation, it is not strictly required for proper synchronization. Any REFCLK/SYNC frequency that satisfies the following equations is acceptable:

$$f_{SYNC} = f_{DACCLK}/2^N \text{ and } f_{SYNC} \leq f_{DATA}$$

where $N$ = 1, 2, 3, or 4.

For example, a configuration with 4× interpolation and clock frequencies of $f_{VCO}$ = 1600 MHz, $f_{DACCLK}$ = 800 MHz, and $f_{DATA}$ = 200 MHz, $f_{SYNC}$ = 100 MHz would be a viable solution.

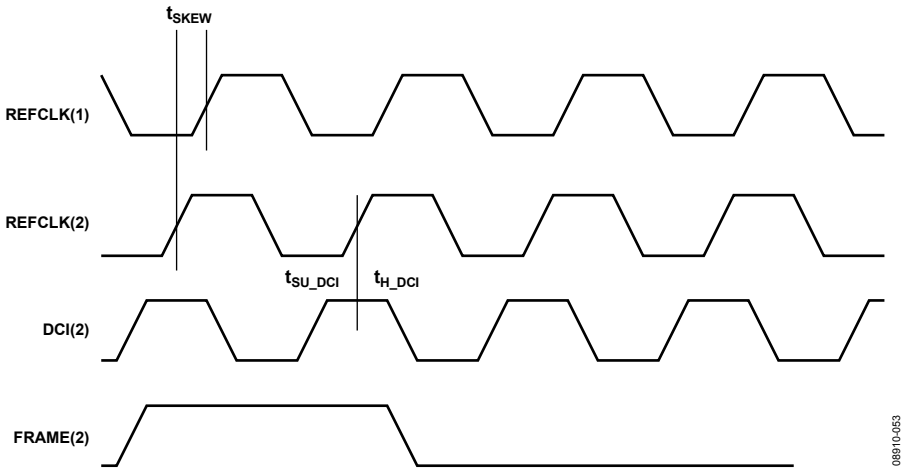



*Figure 54. Timing Diagram Required for Synchronizing Two Devices*

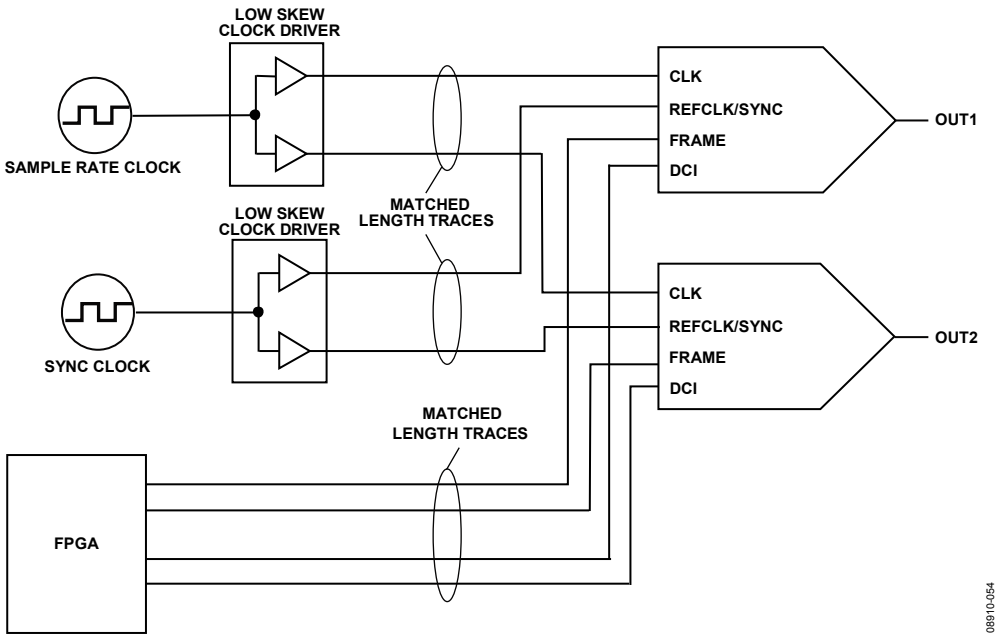



*Figure 55. Typical Circuit Diagram for Synchronizing Devices to a System Clock*

## SYNCHRONIZATION WITH DIRECT CLOCKING

When directly sourcing the DAC sample rate clock to CLK, a separate REFCLK/SYNC input signal is required for synchronization. To synchronize devices, the CLK signals and the REFCLK/SYNC signals must be distributed with low skew to all of the devices being synchronized. This configuration is shown below in Figure 55.

### *Data Rate Mode Synchronization*

The following procedure outlines the steps required to synchronize multiple devices in data rate mode. The procedure assumes that the CLK and REFCLK/SYNC signals are applied to all of the devices. Each individual device must follow the procedure.

The procedure for data rate synchronization when directly sourcing the DAC sampling clock follows:

1. Configure for data rate, periodic synchronization by writing 0xC0 to the sync control register (Register 0x10). Additional synchronization options are available (see the Additional Synchronization Features section).
2. Poll the sync locked bit (Bit 6, Register 0x12) to verify that the device is back-end synchronized. A high level on this bit indicates that the clocks are running with a constant and known phase relative to the sync signal.
3. Reset the FIFO by strobing the FRAME signal for at least the time interval needed to load complete data to the four DACs Resetting the FIFO ensures that the correct data is being read from the FIFO of each of the devices simultaneously. This completes the synchronization procedure, and at this stage, all devices should be synchronized.

To ensure that each of the DACs are updated with the correct data on the same DACCLK edge, two timing relationships must be met on each DAC. DCI (and data) must meet the setup and hold times with respect to the rising edge of CLK, and REFCLK/SYNC must also meet the setup and hold time with respect to the rising edge of CLK. When resetting the FIFO, the FRAME signal must be held high for at least the time interval needed to load complete data to the four DACs (one DCI period for dual-port mode and two DCI periods for single-port or byte mode). When these conditions are met, the outputs of the DACs will be updated within $t_{SKEW}$ + $t_{OUTDLY}$ nanoseconds of each other. A timing diagram that illustrates the timing requirements of the input signals is shown in Figure 56.

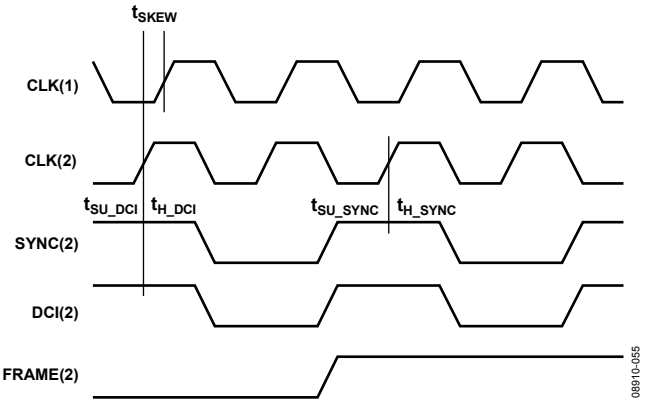



*Figure 56. Synchronization Signal Timing Requirements in Data Rate Mode, 2× Interpolation*

Figure 56 shows the synchronization signal timing with 2× interpolation, so that $f_{DCI} = ½ \times f_{CLK}$. The REFCLK/SYNC input is shown equal to the DCI rate. The maximum frequency at which the device can be resynchronized in data rate mode can be expressed as

$$f_{SYNC} = \frac{f_{DATA}}{2^N}$$

for any positive integer, N.

Generally, for values of N equal to or greater than 3, the FIFO rate synchronization mode is chosen.

### FIFO Rate Mode Synchronization

The following procedure outlines the steps required to synchronize multiple devices in FIFO rate mode. The procedure assumes that the CLK and REFCLK/SYNC signals are applied to all of the devices. Each individual device must follow the procedure.

The procedure for FIFO rate synchronization when directly sourcing the DAC sampling clock follows:

1. Configure for FIFO rate, periodic synchronization by writing 0x80 to the sync control register (Register 0x10). Additional synchronization options are available and are described in the Additional Synchronization Features section.
2. Poll the sync locked bit (Bit 6, Register 0x12) to verify that the device is back-end synchronized. A high level on this bit indicates that the clocks are running with a constant and known phase relative to the sync signal.
3. Reset the FIFO by strobing the FRAME signal high for at least the time interval needed to load complete data to the four DACs. Resetting the FIFO ensures that the correct data is being read from the FIFO of each of the devices simultaneously. This completes the synchronization procedure, and at this stage, all devices should be synchronized.

To ensure that each of the DACs is updated with the correct data on the same DACCLK edge, two timing relationships must be met on each DAC. DCI (and data) must meet the setup and hold times with respect to the rising edge of CLK, and REFCLK/SYNC must also meet the setup and hold time with respect to the rising edge of CLK. When resetting the FIFO, the FRAME signal must be held high for at least the time interval needed to load complete data to the four DACs (one DCI period for dual-port mode, and two DCI periods for single-port or byte mode). When these conditions are met, the outputs of the DACs will be updated within $t_{SKEW}$ + $t_{OUTDLY}$ nanoseconds of each other. A timing diagram that illustrates the timing requirements of the input signals is shown in Figure 57.

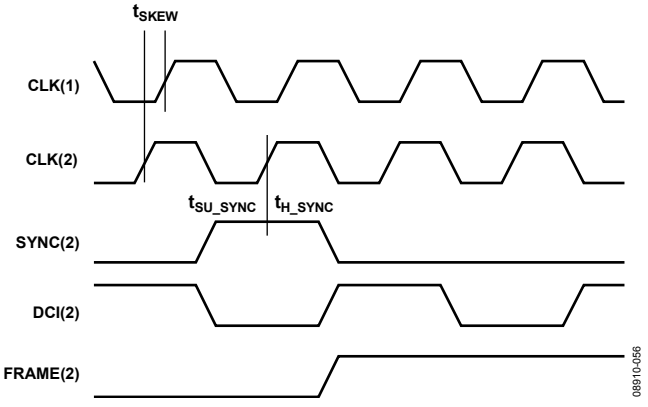



*Figure 57. Synchronization Signal Timing Requirements in FIFO Rate Mode, 2× Interpolation*

Figure 57 shows the synchronization signal timing with 2× interpolation, so that $f_{DCI}$ = ½ × $f_{CLK}$. The REFCLK/SYNC input is shown equal to the FIFO rate. The maximum frequency at which the device can be resynchronized in FIFO rate mode can be expressed as

$$f_{SYNC} = \frac{f_{DATA}}{8 \times 2^N}$$

for any positive integer, N.

## ADDITIONAL SYNCHRONIZATION FEATURES

The synchronization logic incorporates additional features that provide means for querying the status of the synchronization and for improving the robustness of the synchronization. For more information on these features, see the Sync Status Bits section and the Timing Optimization section.

### Sync Status Bits

When the sync locked bit (Bit 6, Register 0x12) is set, it indicates that the synchronization logic has reached alignment. This is determined when the clock generation state machine phase is constant. This takes between (11 + Averaging) × 64 and (11 + Averaging) × 128 DACCLK cycles. This bit may optionally trigger an $\overline{\text{IRQ}}$, as described in the Interrupt Request Operation section.

When the sync lost bit (Bit 7, Register 0x12) is set, it indicates that a previously synchronized device has lost alignment. This bit is latched and remains set until cleared by overwriting the register. This bit may optionally trigger an $\overline{\text{IRQ}}$ as described in the Interrupt Request Operation section.

### Timing Optimization

The REFCLK/SYNC signal is sampled by a version of the DACCLK. If sampling errors are detected, the opposite sampling edge can be selected to improve the sampling point. The sampling edge can be selected by setting Bit 3, Register 0x10 (1 = rising and 0 = falling).

The synchronization logic resynchronizes when a phase change between the REFCLK/SYNC signal and the state of the clock generation state machine exceeds a threshold. To mitigate the effects of jitter and prevent erroneous resynchronizations, the relative phase can be averaged. The amount of averaging is set by the sync averaging bits (Bits[2:0], Register 0x10) and can be set from 1 to 128. The higher the number of averages, the more slowly the device recognizes and resynchronizes to a legitimate phase correction. Generally, the averaging should be made as large as possible while still meeting the allotted resynchronization time interval.

Additional information on synchronization can be found in the AN-1093 Application Note, *Synchronization of Multiple AD9122 TxDAC+ Converters*.

**Table 15. Synchronization Setup and Hold Times**

| Parameter | Min | Max | Unit |
|---|---|---|---|
| $t_{SKEW}$ | $-t_{DACCLK}/2$ | $+t_{DACCLK}/2$ | ps |
| $t_{SU\_SYNC}$ | −100 | | ps |
| $t_{H\_SYNC}$ | +400 | | ps |

# INTERFACE TIMING

The timing diagram for the digital interface port is shown in Figure 59. The sampling point of the data bus nominally occurs 250 ps after each edge of the DCI signal and has an uncertainty of ± 250 ps when the DCI delay is set to 00b (Register 0x72[1:0]), as illustrated by the sampling interval. The data and FRAME signals must be valid throughout this sampling interval. The data and FRAME signals may change at any time between sampling intervals.

The setup ($t_S$) and hold ($t_H$) times with respect to the edges are shown in Figure 59. The minimum setup and hold times are shown in Table 16.

**Table 16. Data Port Setup and Hold Times**

| DCI Delay (Register 0x72, Bits[1:0]) | Minimum Setup Time, $t_S$ (ns) | Minimum Hold Time, $t_H$ (ns) |
|---|---|---|
| 00 | −0.02 | 0.52 |
| 01 | −0.16 | 0.78 |
| 10 | −0.28 | 1.03 |
| 11 | −0.36 | 1.16 |

The data interface timing can be verified by using the SED circuitry. See the Interface Timing Validation section for details.

In data rate mode with synchronization enabled, a second timing constraint between DCI and DACCLK must be met in addition to the DCI-to-data timing shown in Table 17. In data rate mode, only one FIFO slot is being used. The DCI to DACCLK timing restriction is required to prevent data being written to and read from the FIFO slot at the same time. The required timing between DCI and DACCLK is shown in Figure 58.

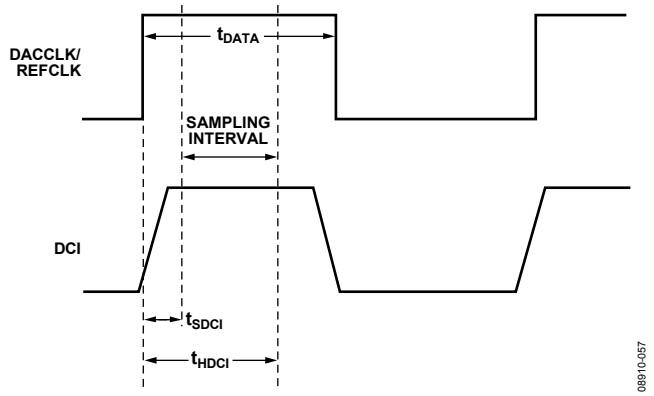



*Figure 58. Timing Diagram for Input Data Port (Data Rate Mode with Sync On)*

**Table 17. DCI to DACCLK Setup and Hold Times vs. DCI Delay Value**

| DCI Delay (Register 0x72,Bits[1:0]) | Minimum Setup Time, $t_{SDCI}$ (ns) | Minimum Hold Time, $t_{HDCI}$ (ns) |
|---|---|---|
| 00 | −0.06 | 0.85 |
| 01 | −0.22 | 1.14 |
| 10 | −0.36 | 1.43 |
| 11 | −0.45 | 1.59 |

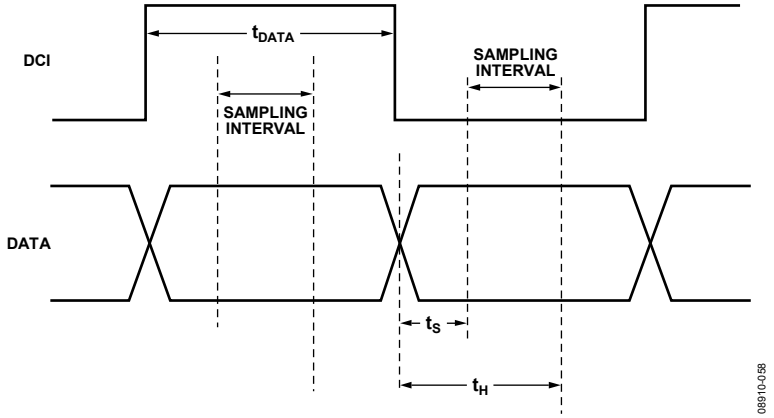



*Figure 59. Timing Diagram for Input Data Ports*

# DIGITAL DATA PATH

The block diagram in Figure 60 shows the functionality of the complex digital data path. The digital processing includes a premodulation block, a programmable complex filter, three half-band interpolation filters with built-in coarse modulation, a quadrature modulator with a fine resolution NCO as well as phase, gain, and offset adjustment blocks.

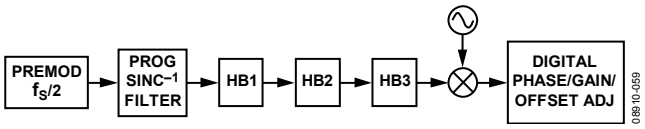



*Figure 60. Block Diagram of Digital Data Path*

There are two complex digital data paths that feed the four DACs. Each digital data path accepts I and Q data streams and processes them as a quadrature data stream, resulting in two quadrature data streams. All of the signal processing blocks can be used when the input data stream is represented as complex data.

The data path can be used to process an input data stream representing four independent real data streams as well; however, the functionality is somewhat restricted. The premodulation block can be used, as well as any of the nonshifted interpolation filter modes.

## PREMODULATION

The half-band interpolation filters have selectable pass bands that allow the center frequencies to be moved in increments of ½ of their input data rate. The premodulation block provides a digital upconversion of the incoming waveform by ½ of the incoming data rate, $f_{DATA}$. Functionally, the premodulation multiplies the incoming data samples alternatively by +1 and −1. This can be used to frequency shift baseband input data to the center of the interpolation filters pass band.

## PROGRAMMABLE INVERSE SINC FILTER

The AD9148 provides a programmable inverse sinc filter to compensate the DAC roll-off over frequency. Because this filter is implemented before the interpolation filter, its coefficients must be changed depending on the interpolation rate and DAC output center frequency.

### *Filter Structure*

The programmable inverse sinc filter is a nine-tap complex FIR filter using complex conjugate coefficients. The z-transfer function is

$$H(z) = \frac{y_I + j \times y_Q}{x_I + j \times x_Q} = H_I + j \times H_Q$$
$$= c_0 + c_1 \times z^{-1} + c_2 \times z^{-2} + c_3 \times z^{-3} + c_4 \times z^{-4}$$
$$+ \bar{c}_3 \times z^{-5} + \bar{c}_2 \times z^{-6} + \bar{c}_1 \times z^{-7} + \bar{c}_0 \times z^{-8}$$

where:

$x_I$ and $x_Q$ are the in-phase (real) and quadrature (imaginary) filter input, respectively.

$y_I$ and $y_Q$ are the in-phase (real) and quadrature (imaginary) filter output, respectively.

$H_I$ and $H_Q$ are the in-phase (real) and quadrature (imaginary) filter coefficients, respectively.

$c_0$, $c_1$, $c_2$, $c_3$, and $c_4$ are the complex filter coefficient, and $\bar{c}_X$ their complex conjugate.

The filter coefficients must be calculated and programmed into the AD9148 registers to perform the operation desired.

### *Filter Implementation*

To perform the complex filtering of the complex input, the filter is divided in four filters working in parallel, two sets of $H_I$ and two sets of $H_Q$ (see Figure 61).

$$y_I + j \times y_Q = (H_I + j \times H_Q) \cdot (x_i + j \times x_Q)$$
$$= H_I \times x_I - H_Q \times x_Q + j \cdot (H_Q \times x_I + H_I \times x_Q)$$

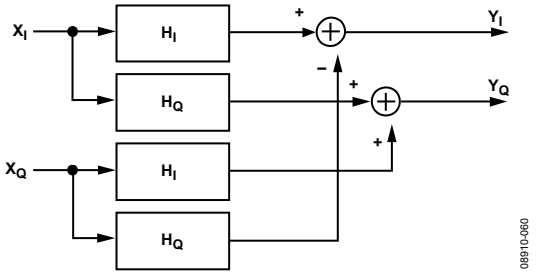



*Figure 61. Complex Filter Implementation*

The coefficients for the filter are stored in SPI Register 0x20 to Register 0x27 in twos-complement format. They have variable length, three bits to 10 bits.

**Table 18. Programmable Inverse Sinc Filter Coefficient Widths and Ranges**

| Coefficient | Width | Minimum | Maximum |
|---|---|---|---|
| $c_0$ in-phase (real) | 3 | 100b | 011b |
| | | −4 | 3 |
| $c_0$ quadrature (imaginary) | 3 | 0100b | 011b |
| | | −4 | 3 |
| $c_1$ in-phase (real) | 4 | 1000b | 0111b |
| | | −8 | 7 |
| $c_1$ quadrature (imaginary) | 4 | 1000b | 0111b |
| | | −8 | 7 |
| $c_2$ in-phase (real) | 5 | 10000b | 01111b |
| | | −16 | 15 |
| $c_2$ quadrature (imaginary) | 5 | 10000b | 01111b |
| | | −16 | 15 |
| $c_3$ in-phase (real) | 7 | 1000000b | 0111111b |
| | | −64 | 63 |
| $c_3$ quadrature (imaginary) | 7 | 1000000b | 0111111b |
| | | −64 | 63 |
| $c_4$ in-phase (real) | 10 | 1000000000b | 0111111111b |
| | | −1024 | 1023 |
| $c_4$ quadrature (imaginary) | 10 | 1000000000b | 0111111111b |
| | | −1024 | 1023 |

The real and imaginary filters are implemented using the structure described in Figure 62 and Figure 63.

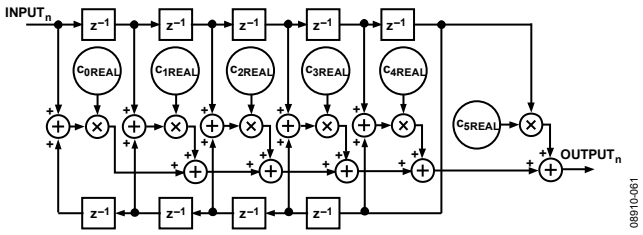



*Figure 62. Real Filter implementation*

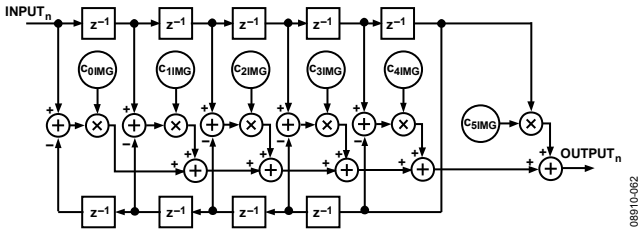



*Figure 63. Imaginary Filter implementation*

The AD9148 evaluation tools provide software that allows for the processing of the filter coefficients based on the DAC sampling frequency, the amount of interpolation used (combination of HB1, HB2, and HB3), and the desired center frequency. This center frequency is limited to

$$[-0.4 \times f_{DAC} + 0.5 \times signalBW, 0.4 \times f_{DAC} - 0.5 \times signalBW]$$

The bandwidth of the inverse sinc filter equals the maximum allowable signal bandwidth of the interpolation filters ($0.8 \times f_{DATA}$).

When there is no interpolation used, the real filter coefficients can be fixed at (no imaginary coefficients)

$C_0 = 2$ ; $C_8 = 2$
$C_1 = -4$ ; $C_7 = -4$
$C_2 = 10$ ; $C_6 = 10$
$C_3 = -35$ ; $C_5 = -35$
$C_4 = 401$

## INTERPOLATION FILTERS

The transmit path contains three interpolation filters. Each of the three interpolation filters provides a 2× increase in output data rate. The filters can be cascaded to provide 2×, 4×, or 8× interpolation ratios. Each of the half-band filter stages offers a different combination of bandwidths and operating modes.

The bandwidth of the three half-band filters with respect to the data rate at the filter input is as follows:

- Bandwidth of HB1 = $0.8 \times f_{IN1}$
- Bandwidth of HB2 = $0.5 \times f_{IN2}$
- Bandwidth of HB3 = $0.4 \times f_{IN3}$

The usable bandwidth is defined as the frequency over which the filters have a pass-band ripple of less than ±0.001 dB and an image rejection of greater than +85 dB. As is discussed in the Half-Band Filter 1 (HB1) section, the image rejection usually sets the usable bandwidth of the filter, not the pass-band flatness.

The half-band filters operate in several modes, providing programmable pass-band center frequencies as well as signal modulation. The HB1 filter has four modes of operation, and the HB2 and HB3 filters each have eight modes of operation.

### *Half-Band Filter 1 (HB1)*

HB1 has four modes of operation, as shown in Figure 64. The shape of the filter response is identical in each of the four modes. The four modes are distinguished by two factors: the filter center frequency and whether the filter modulates the input signal.

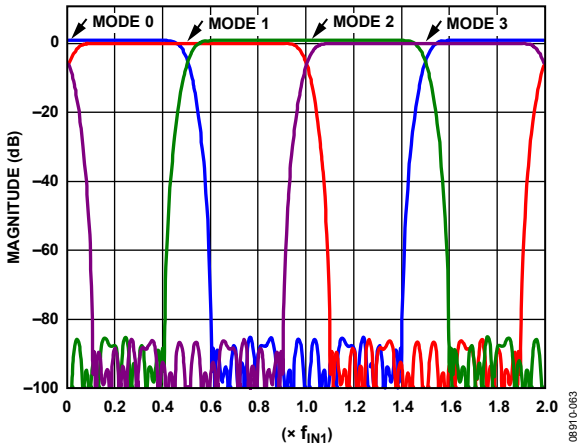



*Figure 64. HB1 Filter Modes*

As is shown in Figure 64, the center frequency in each mode is offset by ½ of the input data rate ($f_{IN1}$) of the filter. Mode 0 and Mode 1 do not modulate the input signal. Mode 2 and Mode 3 modulate the input signal by $f_{IN1}$. When operating in Mode 0 and Mode 2, the I and Q paths operate independently and no mixing of the data between channels occurs. When operating in Mode 1 and Mode 3, mixing of the data between the I and Q paths occurs; therefore, the data input into the filter is assumed complex. Table 19 summarizes the HB1 modes.

**Table 19. 2× Interpolation Filter Modes (Register 0x1C to Register 0x1E)**

| Interpolation Factor | Filter Modes | | | | $f_{CENTER}$ ($f_{DAC}$) |
|---|---|---|---|---|---|
| | Pre-Mod | HB1 | HB2 | HB3 | |
| 2 | 0 | 0 | Off | Off | 0 |
| 2 | 1 | 1 | Off | Off | $f_{DAC}/4$ |
| 2 | 0 | 2 | Off | Off | $f_{DAC}/2$ |
| 2 | 1 | 3 | Off | Off | $-f_{DAC}/4$ |

Figure 65 shows the pass-band filter response for HB1. In most applications, the usable bandwidth of the filter is limited by the image suppression provided by the stop-band rejection and not by the pass-band flatness. Table 20 shows the pass-band flatness and stop-band rejection the HB1 filter supports at different bandwidths.

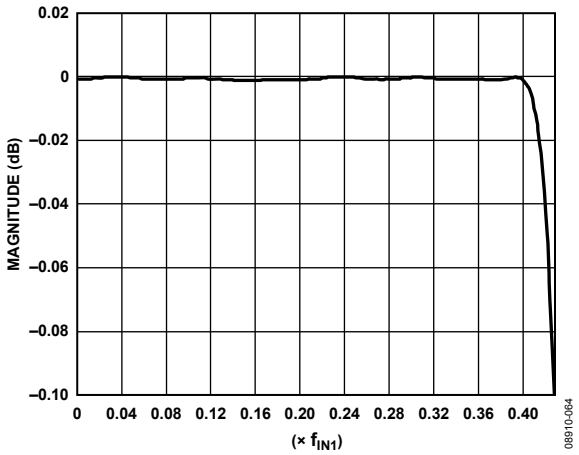



*Figure 65. Pass-Band Detail of HB1*

**Table 20. HB1 Pass-Band and Stop-Band Performance by Bandwidth**

| Bandwidth (% of $f_{IN1}$) | Pass-Band Flatness (dB) | Stop-Band Rejection (dB) |
|---|---|---|
| 80 | 0.001 | 85 |
| 80.4 | 0.0012 | 80 |
| 81.2 | 0.0033 | 70 |
| 82.0 | 0.0076 | 60 |
| 83.6 | 0.0271 | 50 |
| 85.6 | 0.1096 | 40 |

### *Half-Band Filter 2 (HB2)*

HB2 has eight modes of operation, as shown in Figure 66 and Figure 67. The shape of the filter response is identical in each of the eight modes. The eight modes are distinguished by two factors, the filter center frequency and whether the input signal is modulated by the filter.

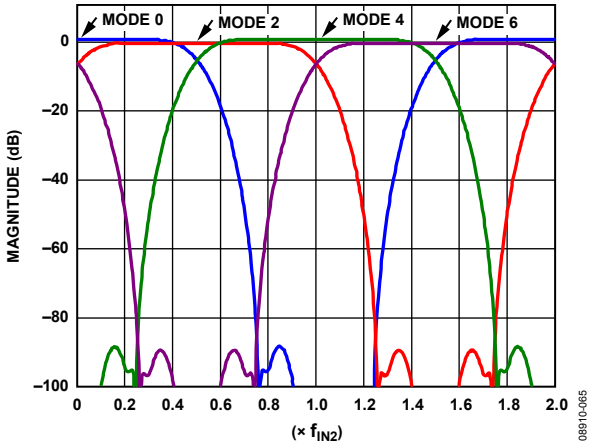



*Figure 66. HB2, Even Filter Modes*

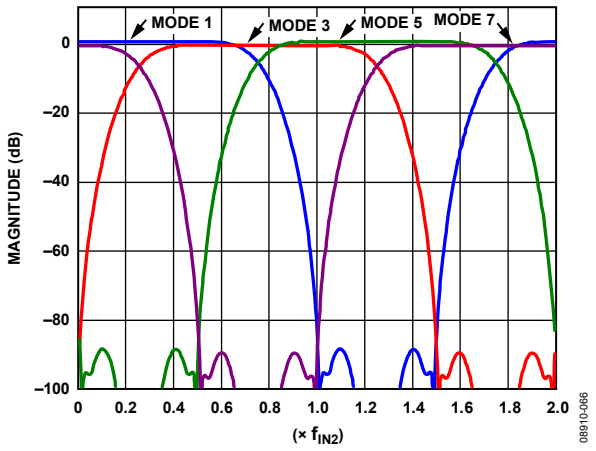



*Figure 67. HB2, Odd Filter Modes*

As shown in Figure 66 and Figure 67, the center frequency in each mode is offset by ¼ of the input data rate ($f_{IN2}$) of the filter. Mode 0 through Mode 3 do not modulate the input signal. Mode 4 through Mode 7 modulate the input signal by $f_{IN2}$. When operating in Mode 0 and Mode 4, the I and Q paths operate independently, and no mixing of the data between channels occurs. When operating in the other six modes, mixing of the data between the I and Q paths occurs; therefore, the data input to the filter is assumed complex. Table 21 summarizes the HB2 modes.

**Table 21. 4× Interpolation Filter Modes (Register 0x1C to Register 0x1E)**

| Interpolation Factor | Filter Modes | | | | $f_{CENTER}$ ($f_{DAC}$) |
|---|---|---|---|---|---|
| | Pre-Mod | HB1 | HB2 | HB3 | |
| 4 | 0 | 0 | 0 | Off | 0 |
| 4 | 1 | 1 | 1 | Off | $f_{DAC}/8$ |
| 4 | 0 | 2 | 2 | Off | $f_{DAC}/4$ |
| 4 | 1 | 3 | 3 | Off | $3f_{DAC}/8$ |
| 4 | 0 | 0 | 4 | Off | $f_{DAC}/2$ |
| 4 | 1 | 1 | 5 | Off | $-3f_{DAC}/8$ |
| 4 | 0 | 2 | 6 | Off | $-f_{DAC}/4$ |
| 4 | 1 | 3 | 7 | Off | $-f_{DAC}/8$ |

Figure 68 shows the pass-band filter response for HB2. In most applications, the usable bandwidth of the filter is limited by the image suppression provided by the stop-band rejection and not by the pass-band flatness. Table 22 shows the pass-band flatness and stop-band rejection the HB2 filter supports at different bandwidths.

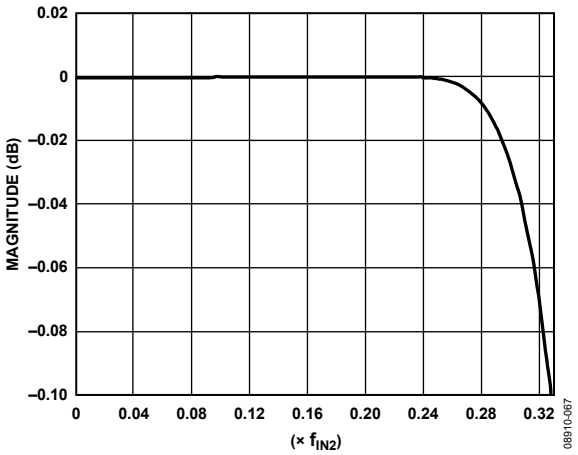



*Figure 68. Pass-Band Detail of HB2*

**Table 22. HB2 Pass-Band and Stop-Band Performance by Bandwidth**

| Bandwidth (% of $f_{IN2}$) | Pass-Band Flatness (dB) | Stop-Band Rejection (dB) |
|---|---|---|
| 50 | 0.001 | 85 |
| 50.8 | 0.0012 | 80 |
| 52.8 | 0.0028 | 70 |
| 56.0 | 0.0089 | 60 |
| 60 | 0.0287 | 50 |
| 64.8 | 0.1877 | 40 |

### *Half-Band Filter 3 (HB3)*

HB3 has eight modes of operation that function the same as HB2. The primary difference between HB2 and HB3 are the filter bandwidths. Table 23 summarizes the filter modes for HB3.

**Table 23. 8× Interpolation Filter Modes (Register 0x1C to Register 0x1E)**

| Interpolation Factor | Filter Modes | | | | $f_{CENTER}$ ($f_{DAC}$) |
|---|---|---|---|---|---|
| | Pre-Mod | HB1 | HB2 | HB3 | |
| 8 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 2 | 2 | 1 | $f_{DAC}/8$ |
| 8 | 0 | 0 | 4 | 2 | $f_{DAC}/4$ |
| 8 | 0 | 2 | 6 | 3 | $3f_{DAC}/8$ |
| 8 | 0 | 0 | 0 | 4 | $f_{DAC}/2$ |
| 8 | 0 | 2 | 2 | 5 | $-3f_{DAC}/8$ |
| 8 | 0 | 0 | 4 | 6 | $-f_{DAC}/4$ |
| 8 | 0 | 2 | 6 | 7 | $-f_{DAC}/8$ |

Figure 69 shows the pass-band filter response for HB3. In most applications, the usable bandwidth of the filter is limited by the image suppression provided by the stop-band rejection and not by the pass-band flatness. Table 24 shows the pass-band flatness and stop-band rejection the HB3 filter supports at different bandwidths.

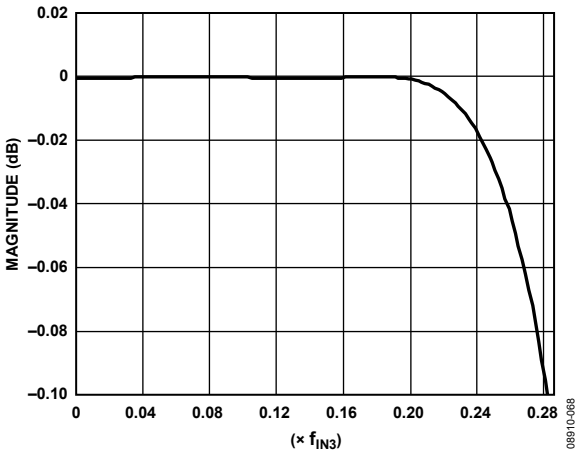



*Figure 69. Pass-Band Detail of HB3*

**Table 24. HB3 Pass-Band and Stop-Band Performance by Bandwidth**

| Bandwidth (% of $f_{IN3}$) | Pass-Band Flatness (dB) | Stop-Band Rejection (dB) |
|---|---|---|
| 40 | 0.001 | 85 |
| 40.8 | 0.0014 | 80 |
| 42.4 | 0.002 | 70 |
| 45.6 | 0.0093 | 60 |
| 49.8 | 0.03 | 50 |
| 55.6 | 0.1 | 40 |

The maximum bandwidth can be achieved if the signal carrier frequency is placed directly at the center of one of the filter pass bands. In this case, the entire quadrature bandwidth of the interpolation filter ($0.8 \times f_{DATA}$) is available. The available signal bandwidth decreases as the carrier frequency of the signal moves away from the center frequency of the filter. The worst-case carrier frequency is one that falls directly between the center frequency of two adjacent filters. Figure 70 shows how the signal bandwidth changes as a function of placement in the spectrum and interpolation rate.

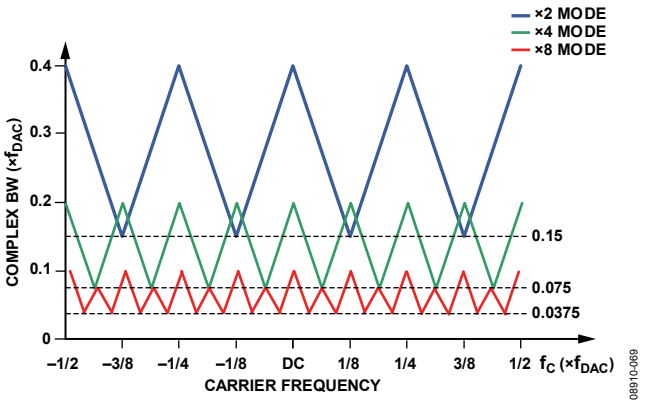



*Figure 70. Complex Signal Bandwidth as a Function of Output Frequency*

## FINE MODULATION

The fine modulation makes use of a numerically controlled oscillator, a phase shifter, and a complex modulator to provide a means for modulating the signal by a programmable carrier signal. A block diagram of the fine modulator is shown in Figure 71. The fine modulator allows the signal to be placed anywhere in the output spectrum with very fine frequency resolution.

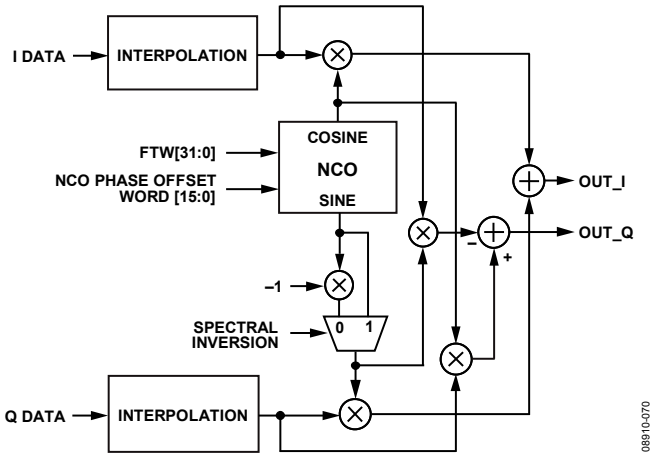



*Figure 71. Fine Modulator Block Diagram*

The quadrature modulator is used to mix the carrier signal generated by the NCO with the I and Q signal. The NCO produces a quadrature carrier signal to translate the input signal to a new center frequency. A complex carrier signal is a pair of sinusoidal waveforms of the same frequency, offset 90° from each other. The frequency of the complex carrier signal is set via the FTW[31:0] value in Register 0x54 through Register 0x57.

The NCO operating frequency, $f_{NCO}$, is at the DAC rate. The frequency of the complex carrier signal can be set from dc up to $f_{DAC}/2$. The frequency tuning word (FTW) is calculated as

$$FTW = \frac{f_{CENTER}}{f_{DAC}} \times 2^{32}$$

The generated quadrature carrier signal is mixed with the I and Q data. The quadrature products are then summed into the I and Q data paths, as shown in Figure 71.

When using the fine modulator, the maximum signal bandwidth of 0.8 × $f_{DATA}$ is always achieved.

### *Updating the Frequency Tuning Word*

The frequency tuning word registers are not updated immediately upon writing as the other configuration registers do. After loading the FTW registers with the desired values, Bit 2 of Register 0x5A must transition from 0 to 1 for the new FTW to take effect.

### *Phase Offset Adjustment*

A 16-bit phase offset may be added to the output of the phase accumulator via the serial port. This static phase adjustment results in an output signal that is offset by a constant angle relative to the nominal signal. This allows the user to phase align the NCO output with some external signal, if necessary. This can be especially useful when NCOs of multiple AD9148s are programmed for synchronization. The phase offset allows for the adjustment of the output timing between the devices. The static phase adjustment is sourced from the NCO Phase Offset[15:0] value located in Register 0x58 and Register 0x59.

## QUADRATURE PHASE CORRECTION

The purpose of the quadrature phase correction block is to enable compensation of the phase imbalance of the analog quadrature modulator following the DAC. If the quadrature modulator has a phase imbalance, the unwanted sideband appears with significant energy. Tuning the quadrature phase adjust value can optimize image rejection in single sideband radios.

Ordinarily, the I and Q channels have an angle of precisely 90° between them. The quadrature phase adjustment is used to change the angle between the I and Q channels. When I Phase Adj, Bits[9:0] (Register 0x28 and Register 0x29), are set to 1000000000b, the I DAC output moves approximately 1.75° away from the Q DAC output, creating an angle of 91.75° between the channels. When I Phase Adj, Bits[9:0] (Register 0x28 and Register 0x29), are set to 0111111111b, the I DAC output moves approximately 1.75° toward the Q DAC output, creating an angle of 88.25° between the channels.

Q Phase Adj, Bits[9:0] (Register 2A and Register 2B), work in a similar fashion. When Q Phase Adj, Bits[9:0] (Register 2A and Register 2B), are set to 1000000000b, the Q DAC output moves approximately 1.75° away from the I DAC output, creating an angle of 91.75° between the channels. When Q Phase Adj[9:0] is set to 0111111111b, the Q DAC output moves approximately 1.75° toward the I DAC output, creating an angle of 88.25° between the channels.

Based on these two endpoints, the combined resolution of the phase compensation register is approximately 3.5°/1024 or 0.00342° per code. When both I Phase Adj, Bits[9:0] (Register 0x28 and Register 0x29), and Q Phase Adj, Bits[9:0] (Register 2A and Register 2B), are used, the full phase adjustment range is ±3.5°.

## DC OFFSET CORRECTION

The dc value of the I data path and the Q data path can be independently controlled by adjusting I DC Offset, Bits[15:0], and Q DC Offset, Bits[15:0], values in Register 0x2C through Register 0x2F. These values are added directly to the data path values. Care should be taken not to overrange the transmitted values.

Figure 72 shows how the DAC offset current varies as a function of I DC Offset, Bits[15:0], and Q DC Offset, Bits[15:0], values. With the digital inputs fixed at midscale (0x000, twos complement data format), Figure 72 shows the nominal $I_{OUTxP}$ and $I_{OUTxN}$ currents as the DC offset value is swept from 0 to 65,535. Because $I_{OUTxP}$ and $I_{OUTxN}$ are complementary current outputs, the sum of $I_{OUTxP}$ and $I_{OUTxN}$ is always 20 mA.

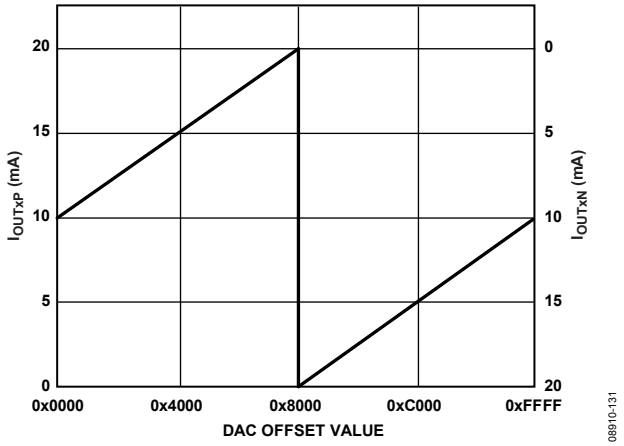



*Figure 72. DAC Output Currents vs. DC Offset Value*

## DIGITAL GAIN CONTROL

The last block in each datapath is an 8-bit scalar (Register 0x50 and Register 0x51) that can be used for digital gain control. The IGain Control, Bits[7:0] (Register 0x50), and QGain control, Bits[7:0] (Register 0x51), values directly scale the samples written to the IDAC and QDAC, respectively. The bit weighting is MSB = $2^1$ and LSB = $2^{-6}$, which yields a multipler range of 0 to 3.984375. The scale factor for each data path is calculated as

$$ScaleFactor = \frac{IGain[7:0]}{64} or \frac{QGain[7:0]}{64}$$

Take care not to overrange the DAC when using a scale factor greater than 1.

# CLOCK GENERATION

## DAC INPUT CLOCK CONFIGURATIONS

The AD9148 DAC sample clock (DACCLK) can be sourced directly or by clock multiplying. Clock multiplying employs the on-chip, phased-locked loop (PLL) that accepts a reference clock (REFCLK_x) operating at a submultiple of the desired DACCLK rate, most commonly the data input frequency. The PLL then multiplies the reference clock up to the desired DACCLK frequency, which can then be used to generate all the internal clocks required by the DAC. The clock multiplier provides a high quality clock that meets the performance requirements of most applications. Using the on-chip clock multiplier removes the burden of generating and distributing the high speed DACCLK.

The second mode bypasses the clock multiplier circuitry and allows DACCLK to be sourced directly through the CLK_x pins. This mode enables the user to source a very high quality clock directly to the DAC core. Sourcing the DACCLK directly through the CLK_x pins may be necessary in demanding applications that require the lowest possible DAC output noise, particularly at higher output frequencies.

## DRIVING THE CLK_x AND REFCLK_x INPUTS

The REFCLK_x and CLK_x differential inputs share similar clock receiver input circuitry. Figure 1 shows a simplified circuit diagram of the input, along with a recommended drive circuit. The on-chip clock receiver has a differential input impedance of about 10 kΩ. It is self-biased to a common-mode voltage of about 1.25 V. The recommended circuit for driving the input is a pair of ac coupling capacitors and a differential 100 Ω termination.

The minimum input drive level to either of the clock inputs is 100 mV ppd. The optimal performance is achieved when the clock input signal is between 500 mV ppd and 1.6 V ppd. Whether using the on-chip clock multiplier or sourcing the DACCLK directly, it is necessary that the input clock signal to the device have low jitter and fast edge rates to optimize the DAC noise performance.

## DIRECT CLOCKING

When a high quality, sample rate clock is connected to the AD9148, it provides the lowest noise spectral density at the DAC outputs. To select the differential CLK inputs as the source for the DAC sampling clock, set the PLL enable bit to 0 (Register 0x0A, Bit 7). Setting this bit to 0 powers down the internal PLL clock multiplier and selects the input from the CLK_x pins as the source for the internal DACCLK.

The device also has duty-cycle correction circuitry and differential input level correction circuitry. Enabling these circuits may provide improved performance in some cases. The control bits for these functions can be found in Register 0x08.

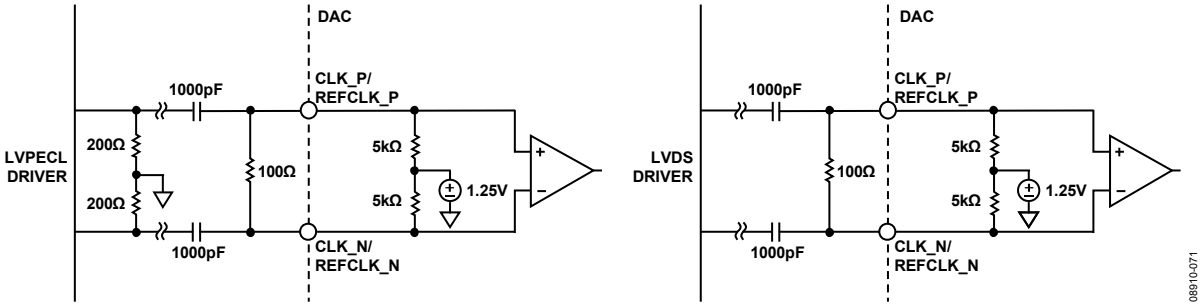



*Figure 73. Clock Receiver Circuitry and Recommended Drive Circuitry using LVPECL (Left) and LVDS (Right)*

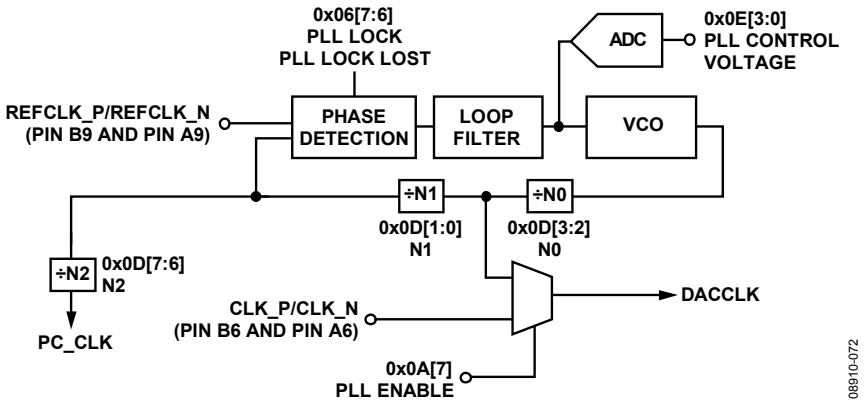



*Figure 74. PLL Clock Multiplication Circuit*

**Table 25. PLL Settings**

| | Address | | |
|---|---|---|---|
| **PLL SPI Control** | **Register** | **Bit** | **Optimal Setting** |
| PLL Loop Bandwidth | 0x0C | [7:5] | 110 |
| PLL Control 1 Register | 0x0C | [4:0] | 01001 |
| PLL Cross Control Enable | 0x0D | [4] | 1 |

## CLOCK MULTIPLICATION

The on-chip PLL clock multiplier circuit can be used to generate the DAC sample rate clock from a lower frequency reference clock. When the PLL clock multiplier is enabled (Register 0x0A[7] = 1), the clock multiplication circuit generates the DAC sample clock from the lower rate REFCLK input. The functional diagram of the clock multiplier is shown in Figure 74.

The clock multiplication circuit operates such that the VCO outputs a frequency, $f_{VCO}$, equal to the REFCLK input signal frequency multiplied by N0 × N1.

$$f_{VCO} = f_{REFCLK} \times (N0 \times N1)$$

The DAC sample clock frequency, $f_{DACCLK}$, is equal to

$$f_{DACCLK} = f_{REFCLK} \times N1$$

The output frequency of the VCO must be chosen to keep $f_{VCO}$ in the optimal operating range of 1.0 GHz to 2.1 GHz. The frequency of the reference clock and the values of N1 and N0 must be chosen so that the desired DACCLK frequency can be synthesized and the VCO output frequency is in the correct range.

### PLL Bias Settings

There are four bias settings for the PLL circuitry that should be programmed to their nominal values. The PLL values shown in Table 25 are the recommended settings for these parameters.

### Configuring the VCO Tuning Band

The PLL VCO has a valid operating range from approximately 1.0 GHz to 2.1 GHz covered in 63 overlapping frequency bands. For any desired VCO output frequency, there may be several valid PLL band select values. The frequency bands of a typical device are shown in Figure 75. Device-to-device variations and operating temperature affect the actual band frequency range. Therefore, it is required that the optimal PLL band select value be determined for each individual device.

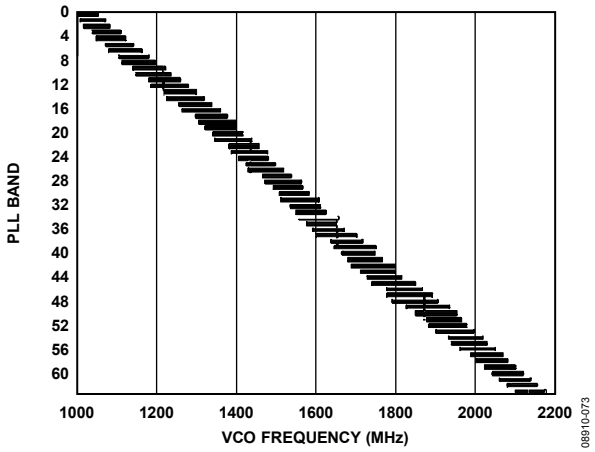



*Figure 75. PLL Lock Range Overtemperature for a Typical Device*

### Automatic VCO Band Select

The device has an automatic VCO band select feature on chip; using this feature is a simple and reliable method for configuring the VCO frequency band. To use the automatic VCO band select feature, enable the PLL by writing 0xC0 to Register 0x0A and enable the auto band select mode by writing 0x80 to Register 0x0A. When this value is written, the device executes an automated routine that determines the optimal VCO band setting for the device. The setting selected by the device ensures that the PLL remains locked over the full −40°C to +85°C operating temperature range of the device without further adjustment. (The PLL remains locked over the full temperature range even if the temperature during initialization is at one of the temperature extremes.)

### *Manual VCO Band Select*

The device also has a manual band select mode that allows the user to select the VCO tuning band. When in manual mode (enabled by setting Bit 6, Register 0x0A to 1), the VCO band is set directly with the value written to the manual VCO band bit enabled (Bits[5:0], Register 0x0A). To properly select the VCO band, complete the following sequence:

1. Put the device in manual band select mode.
2. Sweep the VCO band over a range of bands that results in the PLL being locked.
3. Verify that the PLL is locked and read the VCO control voltage for each band.
4. Select the band that results in the control voltage being closest to the center of the range (that is, 1000). See Table 26 for more details.

The resulting VCO band should be the optimal setting for the device. This band should be written to the manual VCO band register value.

If desired, an indication of where the VCO is within the operating frequency band can be determined by querying the VCO control voltage. Table 26 shows how to interpret the VCO control voltage value.

**Table 26. VCO Control Voltage Range Indications**

| VCO Control Voltage | Indication |
|---|---|
| 1111 | Move to a higher VCO band. |
| 1110 | |
| 1101 | VCO is operating in the higher end of frequency band. |
| 1100 | |
| 1011 | |
| 1010 | |
| 1001 | VCO is operating with an optimal region of the frequency band. |
| 1000 | |
| 0111 | |
| 0110 | |
| 0101 | VCO is operating in the lower end of frequency band. |
| 0100 | |
| 0011 | |
| 0010 | |
| 0001 | Move to a lower VCO band. |
| 0000 | |

# ANALOG OUTPUTS

## TRANSMIT DAC OPERATION

Figure 77 shows a simplified block diagram of one pair of the transmit path DACs. The DAC core consists of a current source array, switch core, digital control logic, and full-scale output current control. The DAC full-scale output current ($I_{OUTFS}$) is nominally 20 mA. The output currents from the IOUTx_P and IOUTx_N pins are complementary, meaning that the sum of the two currents always equals the full-scale current of the DAC. The digital input code to the DAC determines the effective differential current delivered to the load.

The DAC has a 1.2 V band gap reference with an output impedance of 5 kΩ. The reference output voltage appears on the VREF pin. When using the internal reference, the VREF pin should be decoupled to AVSS with a 0.1 µF capacitor. The internal reference should only be used for external circuits that draw dc currents of 2 µA or less. For dynamic loads or static loads greater than 2 µA, the VREF pin should be buffered. If desired, an external reference (between 1.10 V to 1.30 V) can be applied to the VREF pin.

A 10 kΩ external resistor, $R_{SET}$, must be connected from the $\overline{RESET}$ pin to AVSS. This resistor, along with the reference control amplifier, sets up the correct internal bias currents for the DAC. Because the full-scale current is inversely proportional to this resistor, the tolerance of $R_{SET}$ is reflected in the full-scale output amplitude.

The full-scale current can be calculated by

$$I_{OUTFS} = \frac{V_{REF}}{R_{SET}} \times \left(72 + \left(\frac{3}{16} \times DAC\ gain\right)\right)$$

where *DAC gain* is set individually for the I and Q DACs in Register 0x30, Register 0x31, Register 0x34, and Register 0x35, respectively.

For nominal values of VREF (1.2 V), $R_{SET}$ (10 kΩ), and DAC gain (512), the full-scale current of the DAC is typically 20.16 mA. The DAC full-scale current can be adjusted from 8.66 mA to 31.66 mA by setting the DAC gain parameter setting as shown in Figure 76.

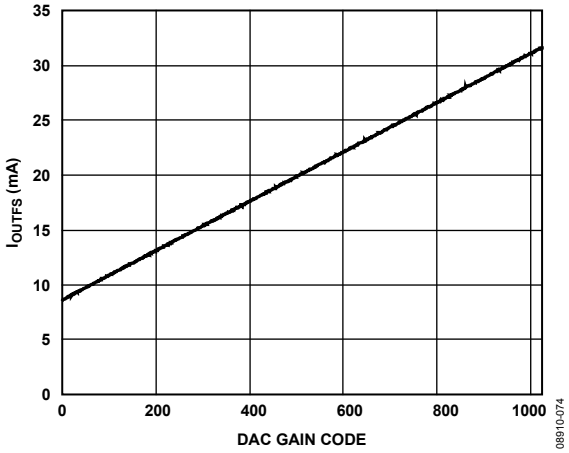



*Figure 76. DAC Full-Scale Current vs. DAC Gain Code*

### *Transmit DAC Transfer Function*

The output currents from the IOUTx_P and IOUTx_N pins are complementary, meaning that the sum of the two currents always equals the full-scale current of the DAC. The digital input code to the DAC determines the effective differential current delivered to the load. IOUTx_P provides the maximum output current when all bits are high. The output currents vs. DACCODE for the DAC outputs are expressed as

$$I_{OUT\_P} = \left[\frac{DACCODE}{2^N}\right] \times I_{OUTFS} \quad (1)$$

$$I_{OUT\_N} = I_{OUTFS} - I_{OUT\_P} \quad (2)$$

where $DACCODE = 0$ to $2^N - 1$.

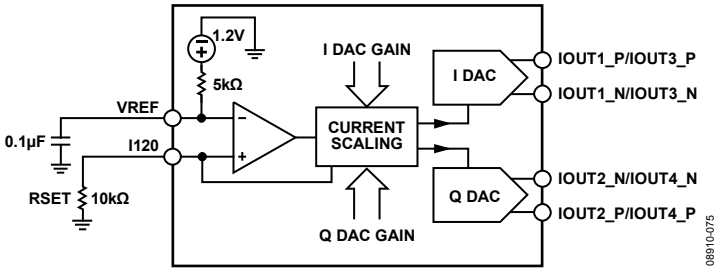



*Figure 77. Simplified Block Diagram of the DAC Core*

### Transmit DAC Output Configurations

The optimum noise and distortion performance of the AD9148 is realized when it is configured for differential operation. The common-mode error sources of the DAC outputs are reduced significantly by the common-mode rejection of a transformer or differential amplifier. These common-mode error sources include even-order distortion products and noise. The enhancement in distortion performance becomes more significant as the frequency content of the reconstructed waveform increases and/or its amplitude increases. This is due to the first-order cancellation of various dynamic common-mode distortion mechanisms, digital feedthrough, and noise.

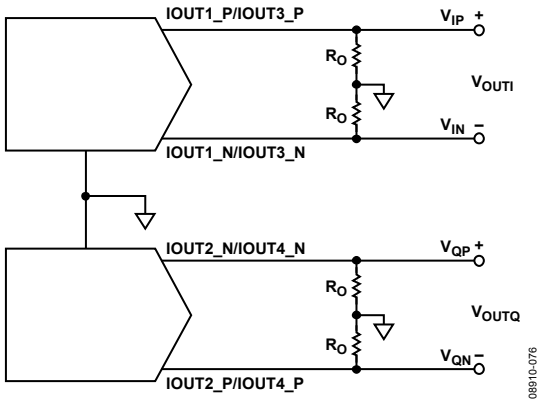



*Figure 78. Basic Transmit DAC Output Circuit*

Figure 78 shows the most basic DAC output circuitry. A pair of resistors, $R_O$, are used to convert each of the complementary output currents to a differential voltage output, $V_{OUT}$. Because the current outputs of the DAC are high impedance, the differential driving point impedance of the DAC outputs, $R_{OUT}$, is equal to $2 \times R_O$. Figure 79 illustrates the output voltage waveforms.

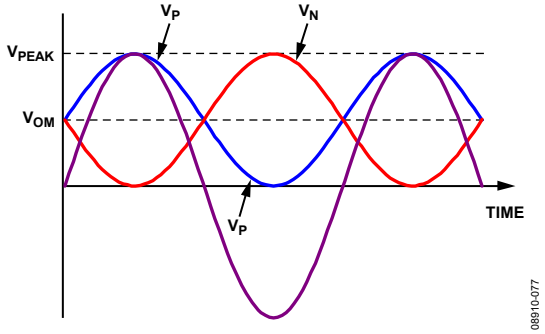



*Figure 79. Voltage Output Waveforms*

The common-mode signal voltage, $V_{CM}$, is calculated by

$$V_{CM} = \frac{I_{FS}}{2} \times R_O$$

The peak output voltage, $V_{PEAK}$, is calculated by

$$V_{PEAK} = I_{FS} \times R_O$$

With this circuit configuration, the single-ended peak voltage is the same as the peak differential output voltage.

### Transmit DAC Linear Output Signal Swing

The DAC outputs have a linear output compliance voltage range of ±1 V that must be adhered to achieve optimum performance. The linear output signal swing is dependent on the full-scale output current, $I_{OUTFS}$, and the common-mode level of the output.

## AUXILIARY DAC OPERATION

The AD9148 has four 10-bit auxiliary DACs (AUX1, AUX2, AUX3, and AUX4). The full-scale output current on these DACs is derived from the 1.2 V band gap reference and external resistor. The gain scale from the reference amplifier current, $I_{REF}$, to the auxiliary DAC reference current is 16.67 with the auxiliary DAC gain set to full-scale. This gives a full-scale current of approximately 2 mA for each auxiliary DAC.

The magnitude of the AUX1 DAC current is controlled via Bits[1:0], Register 0x33 (MSBs) and Bits[7:0], Register 0x32 (LSBs) when DAC SPI select = 0 (Bit 4, Register 0x00). The magnitude of the AUX2 DAC current is controlled via Bits[1:0], Register 0x37 (MSBs) and Bits[7:0], Register 0x36 (LSBs) when DAC SPI select = 0 (Bit 4, Register 0x00). Likewise, the magnitudes of AUX3 DAC current and AUX4 DAC current are controlled via Register 0x33 to Register 0x32 and Register 0x37 to Register 0x36, respectively when DAC SPI Select = 1 (Register 0x00, Bit[4]).

The auxiliary DAC structure is shown in Figure 80. There are two output signals on each auxiliary DAC. One signal is P, and the other is N. The auxiliary DAC outputs are not differential. Only one side of the auxiliary DAC (P or N) is active at one time. The inactive side goes into a high impedance state (100 kΩ). Control of the P side and N side for the auxiliary DACs is via Bit 7, Register 0x33 and Bit 7, Register 0x37 (DAC SPI select is 0 to control AUX1 and AUX2, and DAC SPI select is 1 to control AUX3 and AUX4).

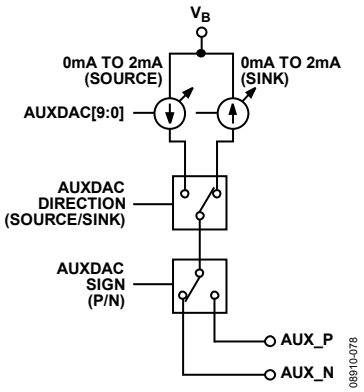



*Figure 80. Auxiliary DAC Structure*

In addition, the P or N output can act as a current source or a current sink. When sourcing current, the output compliance voltage is 0 V to 1.6 V. When sinking current, the output compliance voltage is 0.8 V to 1.6 V. The auxiliary DAC current direction is programmable via Bit 6, Register 0x33 and Bit 6, Register 0x37 (DAC SPI select is 0 to control AUX1 and AUX2, and DAC SPI select is 1 to control AUX3 and AUX4). The choice of sinking or sourcing should be made at circuit design time. There is no advantage to switching between sourcing and sinking current after the circuit is in place.

These auxiliary DACs can be used for local oscillator (LO) cancellation when the DAC output is followed by a quadrature modulator. More information and example application circuits are given in the Interfacing to Modulators section.

## INTERFACING TO MODULATORS

The AD9148 interfaces to the ADL537x family with a minimal number of components. An example of the recommended interface circuitry is shown in Figure 81.

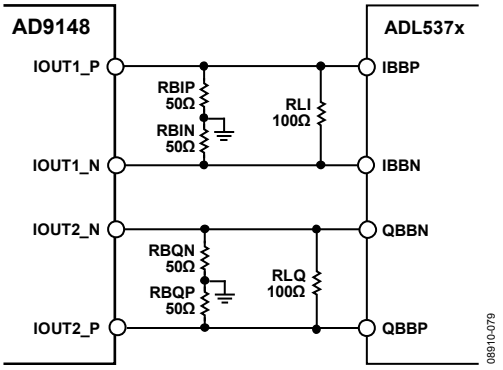



*Figure 81. Typical Interface Circuitry Between the AD9148 and ADL537x Family of Modulators*

The baseband inputs of the ADL537x family require a dc bias of 500 mV. The nominal midscale output current on each output of the DAC is 10 mA (1/2 the full-scale current). Therefore, a single 50 Ω resistor to ground from each of the DAC outputs results in the desired 500 mV dc common-mode bias for the inputs to the ADL537x. The signal level can be reduced by the addition of the load resistor in parallel with the modulator inputs (RLI, RLQ). The peak-to-peak voltage swing of the transmitted signal is

$$V_{SIGNAL} = I_{FS} \times \frac{[2 \times R_B \times R_L]}{[2 \times R_B + R_L]}$$

### *Baseband Filter Implementation*

Most applications require a baseband anti-imaging filter between the DAC and modulator to filter out Nyquist images and broadband DAC noise. The filter can be inserted between the I-to-V resistors at the DAC output and the signal level setting resistor across the modulator input. Doing this establishes the input and output impedances for the filter.

Figure 83 shows a fifth-order low-pass filter. A common-mode choke is used between the I-to-V resistors and the remainder of the filter. This removes the common-mode signal produced by the DAC and prevents the common-mode signal from being converted to a differential signal, which would appear as unwanted spurious signals in the output spectrum. The common-mode choke or balun may not be needed if the layout between the DAC and IQ modulator is optimized and balanced. Splitting the second filter capacitor into two and grounding the center point creates a common-mode low-pass filter, providing additional common-mode rejection of high frequency signals. A purely differential filter passes common-mode signals.

### *Driving the ADL5375-15 with the AD9148*

The ADL5375-15 requires a 1500 mV dc bias and therefore requires a slightly more complex interface than most other Analog Devices, Inc., modulators. It is necessary to level shift the DAC output from a 500 mV dc bias to the 1500 mV dc bias that the ADL5375-15 requires. Level shifting can be achieved with a purely passive network, as shown in Figure 82. In this network, the dc bias of the DAC remains at 500 mV, while the input to the ADL5375-15 is 1500 mV. Note that this passive level shifting network introduces approximately 2 dB of loss in the ac signal.

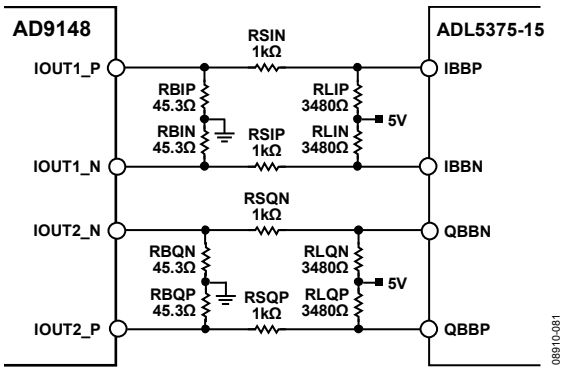



*Figure 82. Passive Level Shifting Network for Biasing the ADL5375-15 from the AD9148*

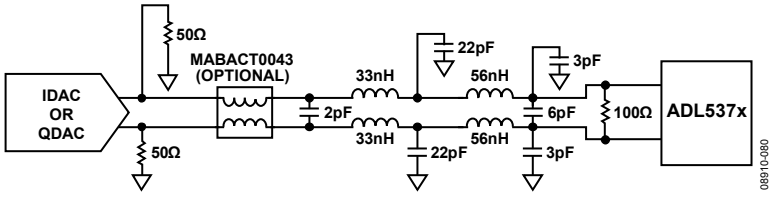



*Figure 83. DAC Modulator Interface with Fifth-Order, Low Pass Filter*

### *Reducing LO Leakage and Unwanted Sidebands*

Analog Devices modulators can introduce unwanted signals at the LO frequency due to dc offset voltages in the I and Q baseband inputs as well as feedthrough paths from the LO input to the output. The LO feedthrough can be nulled by applying the correct dc offset voltages at the DAC output. This can be done either by using the auxiliary DACs (Register 0x32, Register 0x33, Register 0x36, and Register 0x37) or by using the digital dc offset adjustments (Register 0x2C to Register 0x2F). Using the auxiliary DACs has the advantage that none of the main DAC dynamic range is used for performing the dc offset adjustment. The disadvantage is that the common-mode level of the output signal changes as a function of the auxiliary DAC current. The opposite is true when the digital offset adjustment is used.

Good sideband suppression requires both gain and phase matching of the I and Q signals. The phase adjust (Register 0x28 to Register 0x2B) and gain control (Register 0x50 and Register 0x51) registers can be used to calibrate I and Q transmit paths to optimize the sideband suppression. As an alternative to the digital gain scaling, the DAC full-scale output current (Register 0x30, Register 0x31, Register 0x34, and Register 0x35) can also be adjusted to calibrate the I and Q transmit paths; however, changing the DAC full-scale output current affects the common-mode voltage level.

For more information on correcting imperfections in IQ modulators to improve RF signal fidelity, refer to the AN-1039 Application Note.

# DEVICE POWER DISSIPATION

The AD9148 has four supply rails: AVDD33, IOVDD, DVDD18, and CVDD18.

The AVDD33 supply powers the DAC core circuitry. The power dissipation of the AVDD33 supply rail is independent of the digital operating mode and sample rate. The current drawn from the AVDD33 supply rail is typically 98 mA (320 mW) when the full-scale current of the four main DACs (DAC 1, DAC 2, DAC 3, and DAC 4) is set to the nominal value of 20 mA. Changing the full-scale current directly impacts the supply current drawn from the AVDD33 rail. For example, if the full-scale current of the four main DACs is changed to 10 mA, the AVDD33 supply current drops by 40 mA to 58 mA.

The IOVDD voltage supplies the serial port I/O pins (SCLK, SDIO, SDO, CSB, TCK, TDI, TDO, TMS), the $\overline{\text{RESET}}$ pin, and the $\overline{\text{IRQ}}$ pin. The voltage applied to the IOVDD pin can range from 1.8 V to 3.3 V. The current drawn by the IOVDD supply pin is typically 1 mA.

The DVDD18 supply powers all of the digital signal processing blocks of the device. The power consumption from this supply is a function of which digital blocks are enabled and the frequency at which the device is operating.

The CVDD18 supply powers the clock receiver and clock distribution circuitry. The power consumption from this supply varies directly with the operating frequency of the device. CVDD18 also powers the PLL. The power dissipation of the PLL is typically 80 mW.

Figure 84 to Figure 89 detail the power dissipation of the AD9148 under a variety of operating conditions. All of the graphs are taken with data being supplied to all four DACs. The power consumption of the device does not vary significantly with changes in the coarse modulation mode selected or analog output frequency. Graphs of the total power dissipation are shown along with the power dissipation of the DVDD18 and CVDD18 supplies.

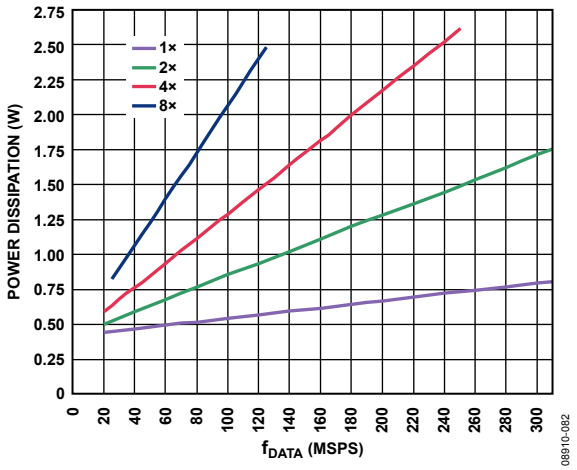



*Figure 84. Total Power Dissipation vs. $f_{DATA}$ with Coarse Modulation, PLL, and Inverse Sinc Filter Disabled*

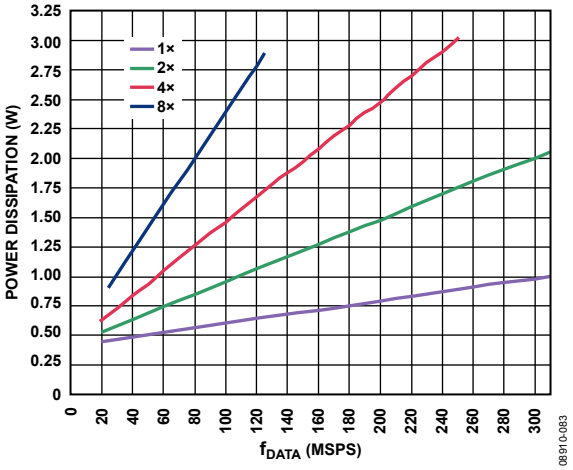



*Figure 85. Total Power Dissipation vs. $f_{DATA}$ with Fine Modulation, PLL, and Inverse Sinc Filter Disabled*

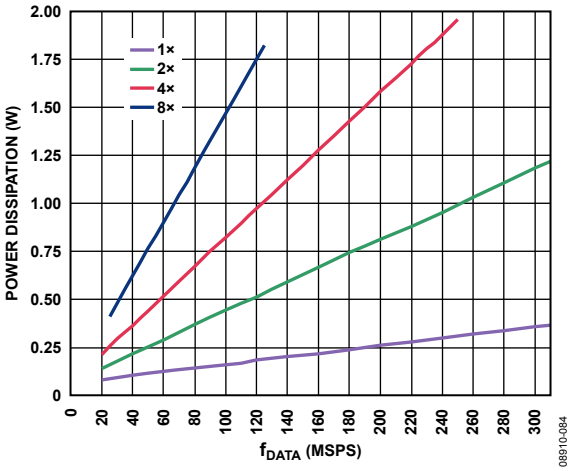



*Figure 86. DVDD18 Power Dissipation vs. $f_{DATA}$ with Coarse Modulation, PLL, and Inverse Sinc Filter Disabled*

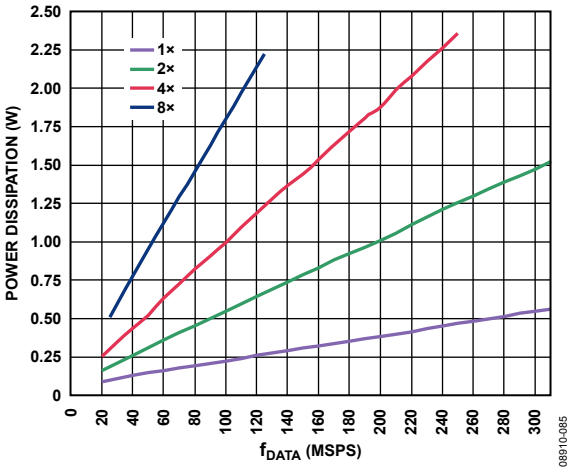



*Figure 87. DVDD18 Power Dissipation vs. $f_{DATA}$ with Fine Modulation, PLL, and Inverse Sinc Filter Disabled*

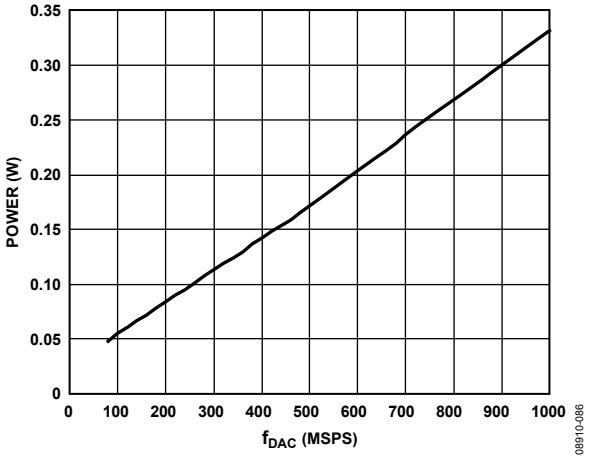



Figure 88. CVDD18 Power Dissipation vs. $f_{DAC}$, PLL Disabled

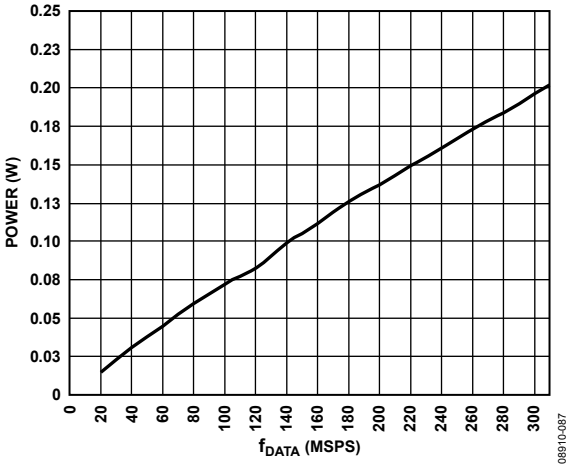



Figure 89. DVDD18 Power Dissipation vs. $f_{DATA}$ Due to Inverse Sinc Filter

# TEMPERATURE SENSOR

The AD9148 has a diode-based temperature sensor for measuring the temperature of the die. The temperature reading is accessed by Register 0x5E and Register 0x5F. The temperature of the die can be calculated as

$$T_{DIE} = \frac{(DieTemp[15:0] - 13{,}700)}{130}$$

where $T_{DIE}$ is the die temperature in degrees Celsius. The temperature accuracy is ±5°C typical over the +85°C to −35°C range. A typical plot of the AD9148 die temperature vs. die temperature code readback is shown in Figure 90.

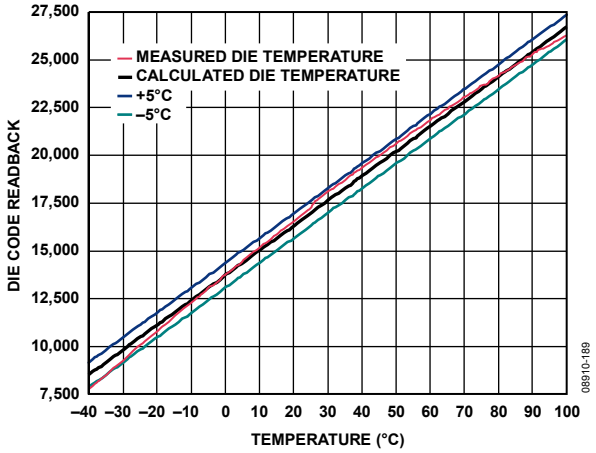



*Figure 90. Die Temperature vs. Die Temperature Code Readback*

Estimates of the ambient temperature can be made if the power dissipation of the device is known. For example, if the device power dissipation is 800 mW and the measured die temperature is 50°C, then the ambient temperature can be calculated as

$$T_A = T_{DIE} - P_D \times T_{JA} = 50 - 0.8 \times 18 = 35.6°C$$

where:
$T_A$ is the ambient temperature in degrees Celsius.
$T_{JA}$ is the thermal resistance from junction to ambient of the AD9148 as shown in Table 7.

To use the temperature sensor, it must be enabled by setting Bit 0, Register 0x5C to 0. Before the temperature sensor data can be read back, it must be latched by toggling Bit 1, Register 0x5C from 0 to 1. In addition, to get accurate readings, the die temperature control register (Register 0x5D) should be set to 0x02.

# INTERRUPT REQUEST OPERATION

The AD9148 provides an interrupt request output signal (Pin H4, $\overline{\text{IRQ}}$) that can be used to notify an external host processor of significant device events. Upon assertion of the interrupt, the device should be queried to determine the precise event that occurred. The $\overline{\text{IRQ}}$ pin is an open-drain, active low output. Pull the $\overline{\text{IRQ}}$ pin high external to the device. This pin may be tied to the interrupt pins of other devices with open-drain outputs to wired-OR these pins together.

Ten different event flags provide visibility into the device. These 10 flags are located in the two event flag registers (Register 0x06 and Register 0x07). The behavior of each of the event flags is independently selected in the interrupt enable registers (Register 0x04 and Register 0x05). When the flag interrupt enable is active, the event flag latches and triggers an external interrupt. When the flag interrupt is disabled, the event flag simply monitors the source signal, and the external $\overline{\text{IRQ}}$ remains inactive.

Figure 91 shows the $\overline{\text{IRQ}}$-related circuitry. Figure 91 shows how the event flag signals propagate to the $\overline{\text{IRQ}}$ output. The interupt_enable signal represents one bit from the interrupt enable register. The event_flag signal represents one bit from the event flag register. The event_flag_source signal represents one of the device signals that can be monitored such as the PLL_locked signal from the PLL phase detector or the FIFO Warning 1 signal from the FIFO controller.

When an interrupt enable bit is set high, the corresponding event flag bit reflects a positively tripped (that is, latched on the rising edge of the event_flag_source version of the event_flag_source signal. This signal also asserts the external $\overline{\text{IRQ}}$. When an interrupt enable bit is set low, the event flag bit reflects the current status of the event_flag_source signal, and the event flag has no effect on the external $\overline{\text{IRQ}}$.

The latched version of an event flag (the interupt_source signal) can be cleared in two ways. The recommended way is by writing 1 to the corresponding event flag bit. A hardware or software reset also clears the interupt_source.

## INTERRUPT SERVICE ROUTINE

Interrupt request management starts by selecting the set of event flags that require host intervention or monitoring. Those events that require host action should be enabled so that the host is notified when they occur. For events requiring host intervention, upon $\overline{\text{IRQ}}$ activation, run the following routine to clear an interrupt request:

- Read the status of the event flag bits that are being monitored.
- Set the interupt enable bit low so that the unlatched event_flag_source can be monitored directly.
- Perform any actions that may be required to quiet the event_source_flag. In many cases, no specific actions may be required.
- Read the event flag to verify that the actions taken have quieted the event_flag_source.
- Clear the interrupt by writing 1 to the event flag bit.
- Set the interrupt enable bits of the events to be monitored.

Noted that some of the event_flag_source signals are latched signals. These are cleared by writing to the corresponding event flag bit. Details of each of the event flags can be found in Table 12.

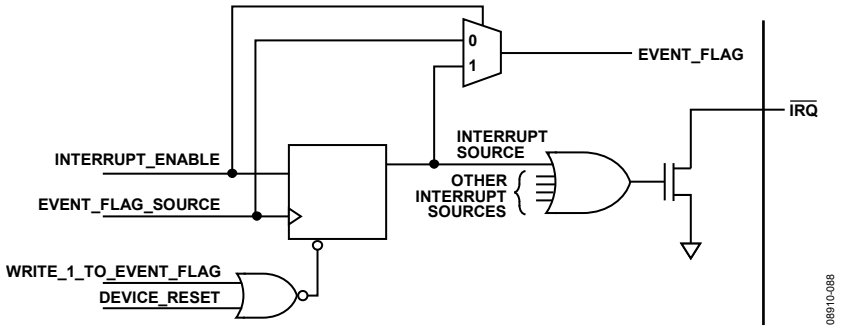



*Figure 91. Simplified Schematic of $\overline{\text{IRQ}}$ Circuitry*

# INTERFACE TIMING VALIDATION

The AD9148 provides on-chip sample error detection (SED) circuitry that simplifies verification of the input data interface. The SED compares the input data samples captured at the digital input pins with a set of comparison values. The comparison values are loaded into registers through the SPI port. Differences between the captured values and the comparison values are detected and stored. Options are available for customizing SED test sequencing and error handling.

## SED OPERATION

The SED circuitry operates on two data sets, one for each data port, each made up of four 16-bit input words, denoted as S0, S1, S2, and S3. To properly align the input samples, the first data-word (that is, S0) is indicated by asserting FRAME for at least one complete input sample.

Figure 92 shows the input timing of the interface for each port. The FRAME signal can be issued once at the start of the data transmission, or it can be asserted repeatedly at intervals coinciding with the S0 and S1 data-words.

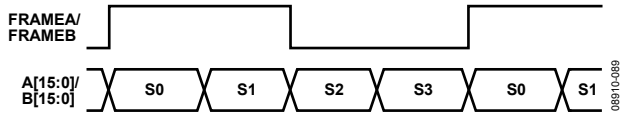



*Figure 92. Timing Diagram of Extended FRAME Signal Required to Align Input Data for SED*

The SED has five flag bits (Register 0x40, Bit 0, Bit 1, Bit 2, Bit 5 and Bit 6) that indicate the results of the input sample comparisons. The sample error detected bit (Bit 5, Register 0x40 for Port A and Bit 6, Register 0x40 for Port B) is set when an error is detected and remains set until cleared. The SED also provides registers that indicate which input data bits experienced errors (Register 0x41 through Register 0x44). These bits are latched and indicate the accumulated errors detected until cleared.

The autoclear mode has two effects: it activates the compare fail bits and the compare pass bit (Register 0x40, Bit 2, Bit 1, and Bit 0) and changes the behavior of Register 0x41 through Register 0x44. The compare pass bit sets if the last comparison indicated that the sample was error free. The compare fail bit sets if an error is detected. The compare fail bit is cleared automatically by the reception of eight consecutive error-free comparisons. When autoclear mode is enabled (Bit 3, Register 0x40), Register 0x41 through Register 0x44 accumulate errors as previously described but reset to all 0s after eight consecutive error-free sample comparisons are made.

The sample error, compare pass, and compare fail flags can be configured to trigger an $\overline{\text{IRQ}}$ when active, if desired. This is done by enabling the appropriate bits in the event flag register (Register 0x07).

## SED EXAMPLE

### *Normal Operation*

The following example illustrates the SED configuration for continuously monitoring the input data and assertion of an $\overline{\text{IRQ}}$ when a single error is detected.

1. Write to the following registers to enable the SED and load the comparison values:
   Register 0x40 → 0x80
   Register 0x00[4] → 0 (to configure Port A SED)
   Register 0x38 → S0[7:0]
   Register 0x39 → S0[15:8]
   Register 0x3A → S1[7:0]
   Register 0x3B → S1[15:8]
   Register 0x3C → S2[7:0]
   Register 0x3D → S2[15:8]
   Register 0x3E → S3[7:0]
   Register 0x3F → S3[15:8]
   Register 0x00[4] → 1 (to configure Port B SED)
   Register 0x38 → S0[7:0]
   Register 0x39 → S0[15:8]
   Register 0x3A → S1[7:0]
   Register 0x3B → S1[15:8]
   Register 0x3C → S2[7:0]
   Register 0x3D → S2[15:8]
   Register 0x3E → S3[7:0]
   Register 0x3F → S3[15:8]
   Comparison values can be chosen arbitrarily; however, choosing values that require frequent bit toggling provides the most robust test.
2. Enable the SED error detect flag to assert the $\overline{\text{IRQ}}$ pin.
   Register 0x05 → 0x04
3. Begin transmitting the input data pattern.

If $\overline{\text{IRQ}}$ is asserted, read Register 0x40 and Register 0x41 through Register 0x44 with Bit 4, Register 0x00 = 0 for Port A and with Bit 4, Register 0x00 = 1 for Port B, to verify that a SED error was detected, and determine which input bits were in error. The bits in Register 0x41 through Register 0x44 are latched; therefore, the bits indicate any errors that occurred on those bits throughout the test and not just the errors that caused the error detected flag to be set.

Note that the FRAME signal is not required during normal operation when the device is configured for dual-port mode. To enable the alignment of the S0 sample as previously described requires the use of both the FRAMEA and FRAMEB signals.

The timing diagrams for single-port and byte modes are the same as during normal operation and are shown in Figure 47 and Figure 48, respectively. For single-port and byte mode, only FRAMEA and the IRQs for Port A should be used. The FRAMEA rising edge should always be aligned with the first sample of the data transmission. There should not be another rising edge until four complete words of data are received. This means four data samples for dual-port mode and eight data samples for single-port and byte modes.

# EXAMPLE START-UP ROUTINE

To ensure reliable start-up of the AD9148, certain sequences should be followed. An example start-up routine using the following device configuration is used for this example.

- $f_{DATA}$ = 122.88 MSPS
- Interpolation = 4×, using HB1 = ’00’ and HB2 = ’000’
- Input data = baseband data
- Dual port mode with 1 DCI
- $f_{OUT}$ = 140 MHz
- $f_{REFCLK}$ = 122.88 MHz
- PLL = enabled
- Fine NCO = enabled
- Inverse sinc filter = disabled
- Synchronization = enabled

## DERIVED PLL SETTINGS

The following PLL settings can be derived from the device configuration:

- $f_{DACCLK} = f_{DATA} \times$ Interpolation = 491.52 MHz
- $f_{VCO} = 4 \times f_{DACCLK}$ = 1966.08 MHz (1 GHz < $f_{VCO}$ < 2 GHz)
- $N1 = f_{DACCLK}/f_{REFCLK} = 4$
- $N0 = f_{VCO}/f_{DACCLK} = 4$

## DERIVED NCO SETTINGS

The following NCO settings can be derived from the device configuration:

- $f_{OUT}$ = 140 MHz
- $f_{DACCLK} = f_{DATA} \times$ Interpolation = 491.52 MHz
- FTW = $140/(491.52) \times 2^{32}$ = 0x48, EAAAAA

## START-UP SEQUENCE

The power clock and register write sequencing for reliable device start-up follows:

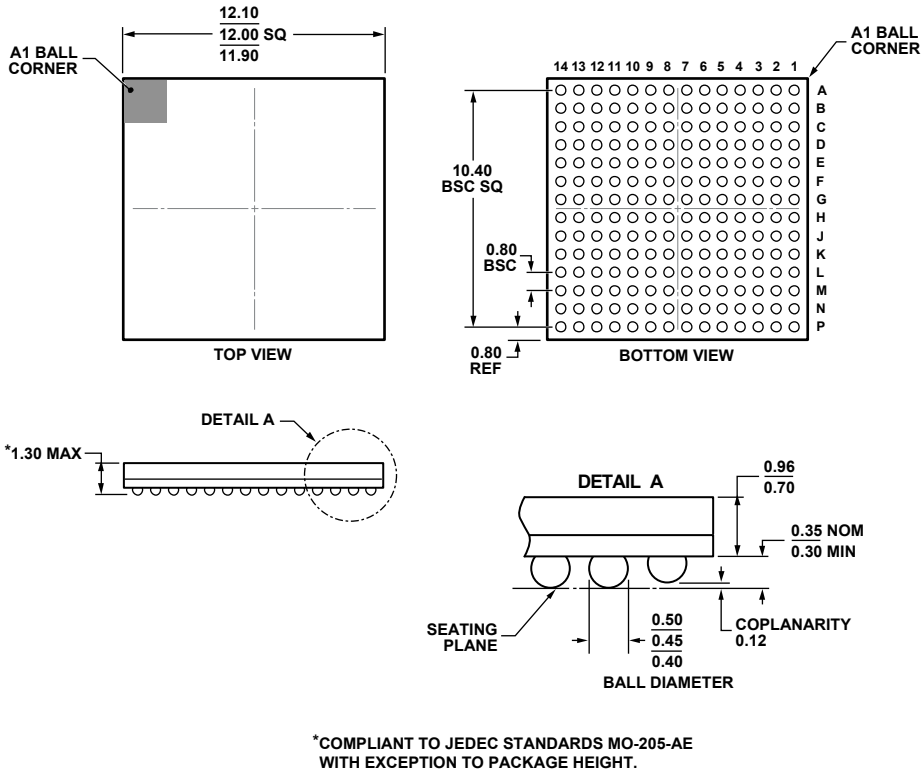
- Power up the device (no specific power supply sequence is required)
- Apply a stable REFCLK input signal.
- Apply a stable DCI input signal.
- Issue a hardware reset (optional)
- Configure device registers with the following write sequence:
  - 0x0C → 0xC9
  - 0x0D → 0xD9
  - 0x0A → 0xC0
  - 0x0A → 0x80
  - 0x10 → 0x48
  - 0x14 → 0x40
  - 0x17 → 0x08
  - 0x17 → 0x00
  - 0x19 → 0x08
  - 0x19 → 0x00
  - 0x1C → 0x40
  - 0x1D → 0x00
  - 0x1E → 0x01
  - 0x54 → 0xAA
  - 0x55 → 0xAA
  - 0x56 → 0xEA
  - 0x57 → 0x48
  - 0x5A → 0x01
  - 0x5A → 0x00

## DEVICE VERIFICATION SEQUENCE

The following device polling can be conducted to verify that the device is working properly:

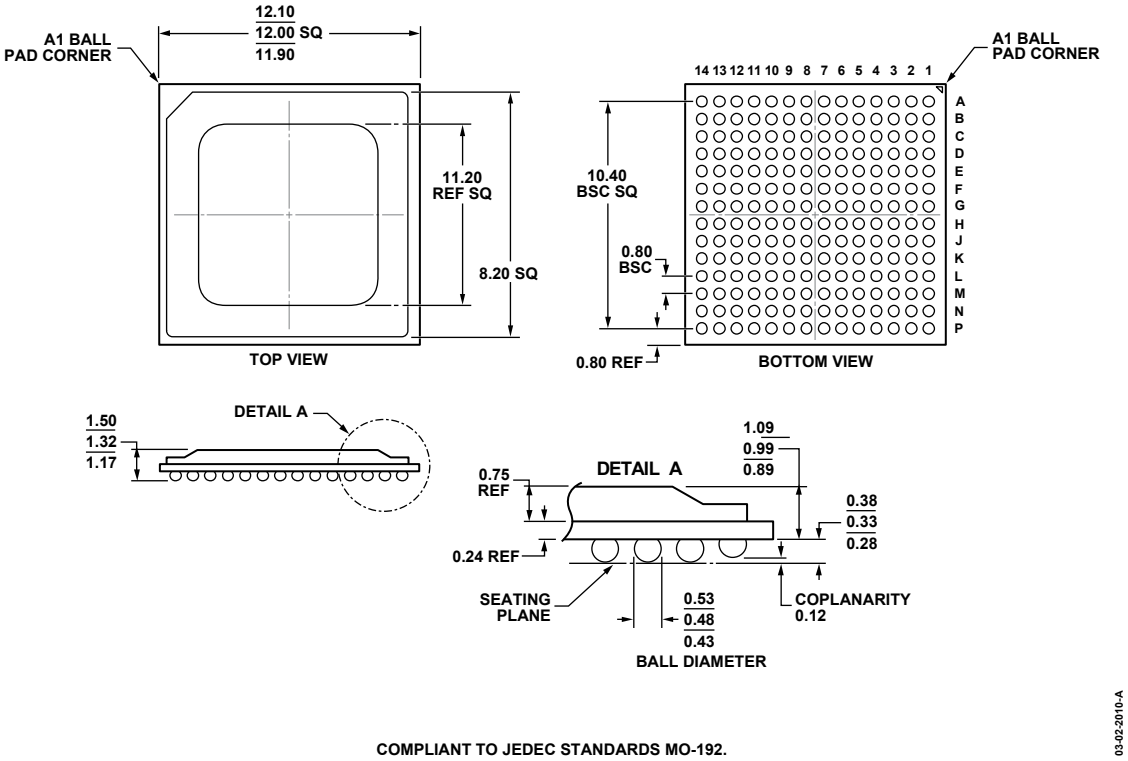
- Read 0x06, Expect Bit 7 = 0, Bit 6 = 1, Bit 5 = 0, Bit 4 = 1, Bit 2 = 1
- Read 0x12, Expect Bit 6 = 1
- Read 0x18, Expect 0x0F (0x07 is also normal)
- Read 0x1A, Expect 0x0F (0x07 is also normal)

# OUTLINE DIMENSIONS




*COMPLIANT TO JEDEC STANDARDS MO-205-AE WITH EXCEPTION TO PACKAGE HEIGHT.

*Figure 93. 196-Ball Chip Scale Package, Ball Grid Array [CSP_BGA]*
*BC-196-7*
*Dimensions shown in millimeters*




COMPLIANT TO JEDEC STANDARDS MO-192.

*Figure 94. 196-Ball Ball Grid Array, Thermally Enhanced [BGA_ED]*
*BP-196-1*
*Dimensions shown in millimeters*

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Package Option |
|---|---|---|---|
| AD9148BBCZ | −40°C to +85°C | 196-Ball Chip Scale Package Ball Grid Array [CSP_BGA] | BC-196-7 |
| AD9148BBCZRL | −40°C to +85°C | 196-Ball Chip Scale Package Ball Grid Array [CSP_BGA] | BC-196-7 |
| AD9148BBPZ | −40°C to +85°C | 196-Ball Ball Grid Array, Thermally Enhanced [BGA_ED] | BP-196-1 |
| AD9148BBPZRL | −40°C to +85°C | 196-Ball Ball Grid Array, Thermally Enhanced [BGA_ED] | BP-196-1 |
| AD9148-EBZ | | DAC Only Evaluation Board [BGA_ED] | BP-196-1 |
| AD9148-M5372-EBZ | | AD9148 + ADL5372 Evaluation Board [BGA_ED] | BP-196-1 |
| AD9148-M5375-EBZ | | AD9148 + ADL5375-0.5 Evaluation Board [BGA_ED] | BP-196-1 |

[1] Z = RoHS Compliant Part.

## NOTES

# NOTES

©2010–2011 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
D08910-0-9/11(A)